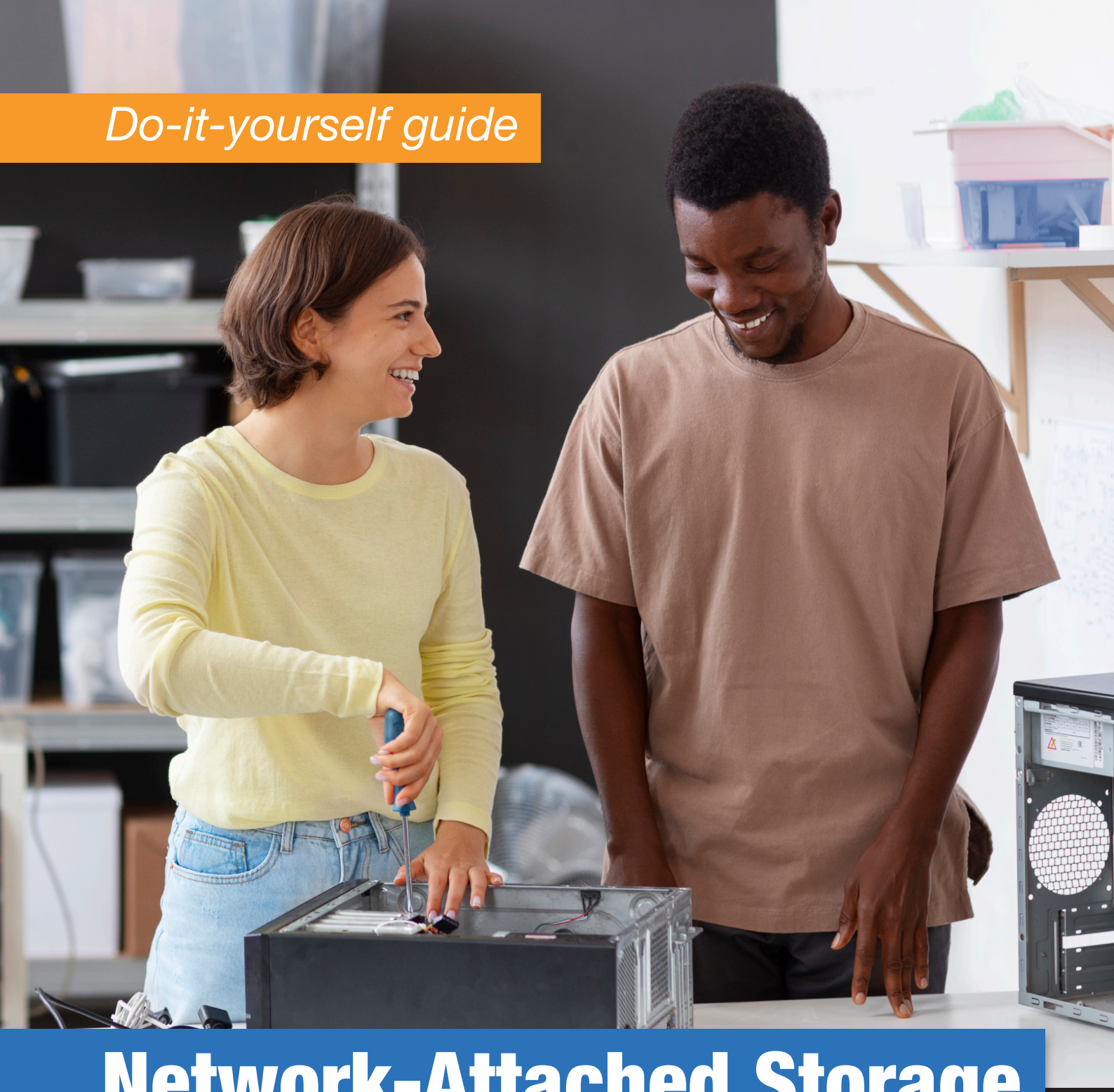


Do-it-yourself guide

Network-Attached Storage

for Data Security and Backups

- A dive into three fundamental concepts
- Tips and explanations that help you make informed decisions
- A step-by-step guide to configure the software
- A practical case study for hands-on learning

SturmFlosseTK | July 2025

Text: CC BY-NC-SA | Images: Indiviual licenses

2st edition (July 2025)
Created by **SturmFlosseTK** (*internet nickname of the author*)

Changes from 1st (May 2025) to 2nd edition:
Implemented (most of the) notes from the first review (see here)

Written content is licensed by the CC BY-NC-SA license.

The visuals used in this manual are either designed by the author itself or designed by the artists at Freepik.com, Flaticon.com and Unsplash.com.

The photo on the cover page was created by Freepik.com.

Some images in this manual are used under separate licenses from Freepik, Flaticon, or Unsplash and are covered differently rather than the core CC BY-NC-SA license.

Rights for every name and logo of any company mentioned belong to it's respective owners.
Every company is only mentioned for informative purposes.

Regarding the content of this manual, everything was either
- teached at the school the author visited,
- researched in the documentation of each operating system or the videos linked,
- or hands-on experience made by the author.

Documentations of each operating system:
- iXsystems (n.d.), TrueNAS Scale documentation, https://www.truenas.com/docs
- Lime Technology, Inc. (n.d.), Unraid documentation, https://wiki.unraid.net
- OpenMediaVault. (n.d.), OpenMediaVault documentation, https://docs.openmediavault.org/en/stable/

Disclaimer

This manual was first created in May 2025 as an academic project. It received the highest score possible and includes a successfull practical implementation with additional tried and tested code published on Github (see case study). It was also well received by colleagues and others.

The author wanted to publish it to the internet so others can build upon the knowledge it provides. This led to an updated version in July 2025 thanks to public feedback.

While its content may become outdated over time, the core concepts remain relevant and the tips and guides provided will still assist you in selecting your own data security and backup solution.

Also, lets get one thing straight: If you are not comfortable with handling individual pc components and you don't want to be in the future, the best option is a prebuilt network-attached storage (NAS) system from manufacturers like Synology, Qnap or UGreen. Choosing such a system will send you on another guided "path" than this manual because this right here is neither a comparison of all NAS devices on the current market nor a guide on how to use them. You can and should read on to get more clearly about the concepts and your personal needs but roughly half of what you will read will not be applicable to a situation with a prebuilt system.

This manual is designed to help you build the most personal NAS for **you** by choosing new or repurposing old spare computer parts. This will save you time and money and you will achieve **true data freedom**!

Please note:
This manual is written as a **„getting started guide“** and will explain many but not all details that can be known about a NAS build!

Additionaly it is for informative purposes only! The author does **not** take any responsibility for any error or damage that happens to any of your systems, data and backups! Please read carefully and do your own additional research online (see the links mentioned on the pages).

Content

Introduction 01
Core concepts 02
The good, the bad, the reality 03
Defining your personal criterias 03
Choosing appropriate hardware 08
- Central Processing Unit (CPU) 09
- CPU-Cooler 09
- Random Access Memory (RAM) 09
- Graphics Processing Unit (GPU) 10
- Power Supply (PSU) 10
- Additional fans 10
- Mainboard („Motherboard“) 10
- PC-Case 11
- Storage devices 11

Assembling the hardware 17
Installing the operating system 19
Software concepts 22
- Virtual devices and „RAID“ 22
- Datasets 27
- SMB share 28
- Backups (Rsync, Snapshots, Replication) 28
- Secure shell (SSH) 35
- Remote access 35
- Summary 36

Software configuration 37
- Creating a pool 37
- Creating datasets 40
- Accessing data with SMB-Shares 41
- Creating snapshot tasks 44
- Creating replication tasks 46

Case study 47
Additional considerations 51
- Security enhancements 51
- Setting up alerts 51
- Creating a plan for disaster 52
- Documentation 52

Final words 53

Introduction

Welcome to the digital age, a place where data is and always will be everywhere we go. In a single day numerous photos are taken and posted online, teachers and pupils are creating and sharing notes and businesses use their own and the data of others to improve their daily workflows.

But is all of this data still relevant if it's not reliable or lost in a blink of an eye? How do we keep this data safe, consistent and readily accessible? In todays world the importance of protecting our personal and professional data cannot be overstated.

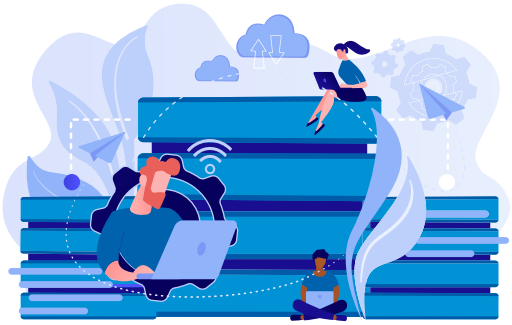
That is why this manual was created. In here we will break down the core concepts of data security in an easy-to-understand language and explore related backup solutions. You will be guided step-by-step on how to build your own NAS system. Whether you are an tech enthusiast or just someone who wants to keep his/her data safe, this manual was created for you.

Let's begin with your journey of building a fortress around your digital life to give you peace of mind in an interconnected world!

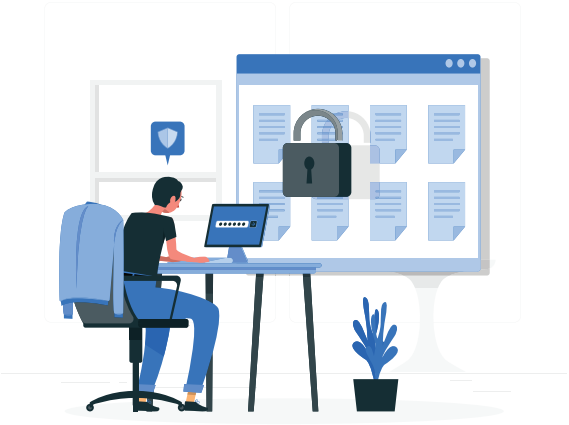
Core concepts

When thinking about data security, we need to know three core concepts. Each concept can be viewed as a core pillar of keeping your data safe.

1. The first concept is "**accessibility**". This refers to the ability to access your data when needed. It ensures that users can retrieve the necessary data without unnecessary delays or obstacles. Accessibility is crucial for efficient workflows and decision-making processes. This also implements the ability to restore deleted data from backups.



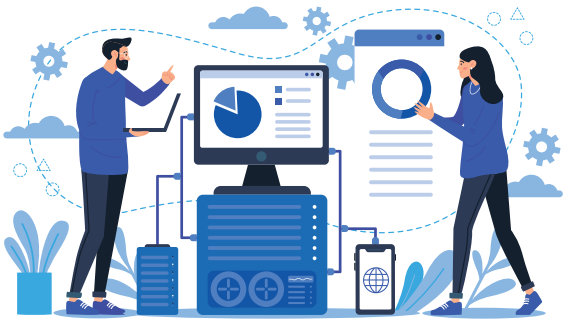
Designed by vectorjuice at Freepik.com



Designed by storyset at Freepik.com

2. But when data is accessible (e.g. to everyone) it is not secure, right? This is where the second concept "**confidentiality**" comes in. This is about protecting data from unauthorized access and ensuring that only authorized individuals can view or modify it. Confidentiality measures are essential to safeguard sensitive information and maintain privacy.

3. By following the two concepts mentioned above we made sure our data is accessible, restorable and that only authorized access is possible. But what if an authorized person accesses your data and implements a mistake - by accident or on purpose? This is why we need to think about "**integrity**", too. This concept is about assuring and maintaining accuracy and consistency of our data over its entire lifecycle. Data integrity ensures that the data is reliable, complete and unaltered during transit and storage, preventing unauthorized changes or data corruption. In simple words: Data integrity is achieved by storing the most recent versions of our files in one safe place rather than spreading it out on multiple systems with many different versions of the same file.



Designed by Freepik at Freepik.com

The good, the bad, the reality

We can achieve a practical implementation of these concepts into our digital life by building a system that takes care of the necessary processes - automatically and on a regular basis. One - and the most likely - option is a network-attached storage (NAS), a system that is connected to a network to offer storage space to one or more authorized clients, keeping data separate from individual devices in a secure place, where automatic backups and safety measurements are a central focus point. Here are the biggest advantages and disadvantages of a NAS system:

Advantages	Disadvantages
Data accessibility	Initial cost for a separate system
Maintaining data integrity	Technical knowledge required
Shared storage for better cooperation	Network required
Easier to backup than individual data of every client	Possibility of a "single point of failure"
Energy Efficiency	
Cost savings in comparison to buying storage for every individual client	
Additional features and services for every client on the network possible	

Defining your personal criterias

Ok, a NAS then... But how?

When starting any project you need to think of your main use case (your biggest focus point). What do you personally want to achieve with this system? What will you be using it for? Why do you need it? Think about this for a bit, fill in your why and then read on.

> I will build a NAS because

On the next page you will find questions that will help you defining your personal criterias more clearly. Note down your thoughts. Afterwards you can refer to these answers while choosing parts for your system.

If you cannot answer every questions right now, don't worry. You can read on and come back to these questions whenever you have your answer.

Question	Your personal answer
How much space do I need (GB / TB)?	
What is more important to me: Fast or big storage?	
How important is data security for me? How critical or sensitive is the data that will be stored on the NAS?	
How many tasks should my NAS fullfill day by day? Only store data? Only acting as a backup server for my main devices? Do i want advanced features like a gaming server?	
Does it need to be scalable? Do I wanna start small and expand my system in the future? What would be a „nice to have“?	
How many users will connect to my NAS?	
What operating systems will be connected to my NAS?	
How will other systems connect to my NAS (via LAN, WLAN or remotely)?	
How often will they access it?	
How many administrators should have access to my NAS?	
What will the physical location of my NAS look like?	
Should my NAS be as silent as possible or will some sound of data processing hard drives and moving fans not be a big of a pain?	
What are my expectations regarding energy consumption (low, high, "whatever it takes")?	
Do I know a specific software that can be run on a NAS and its requirements? If yes, what do I am familiar with?	

One recommendation after answering the questions is to put all of these ideas and informations in a concept by formulating clearly defined sentences to summarize your criterias. You should focus on the "why"-aspect of your criterias.

Here is an example for "low budget, fast storage, accessible from remote locations, one user, one admin":

"*My goal is to design a budget-friendly NAS with SSDs or fast HDDs for optimal storage speed and the ability for remote access because I need my files in remote locations with limited internet speed. I can ensure ease of use for the single user and streamlined administration by the one admin. I will set the focus on a compact, energy-efficient device while delivering reliable performance.*"

Of course this is easier said than done but you can ask a friend or use your favorite AI to help you. For example you could give some questions above to a friend or an AI with the request of giving you three options for every question to choose from. After choosing your favorite options you can ask them to help you summarize your criterias into sentences.

Take your time, write your sentence(s) and then read on.

Now that we transformed an idea into a concept, we need to think about the prerequisites of your personal situation. The following questions are mostly about pc hardware. If you never worked with this, don't worry. Explanations and tips for choosing appropriate hardware will follow in the next chapter.

Question	**Your answer**
What pc components am I comfortable working with?	
For what compenent(s) do i know what to look out for when buying new or used?	

Do I know someone who knows more about „pc stuff“ than me?	
What components do I already have on hand (that you don't need for something else)?	
Do I know what network I am working on? Do I have control (access to the settings) over the network? Where in the network should the NAS be located? Don't worry: If you never modified a network before you probably have one big network at home (in which a NAS will be very easy to implement).	
IMPORTANT: What's my allocated budget? (Most often it is better to set a price range rather than a fixed number)	

Please note that old spare computer parts are most often powerful enough for a basic NAS. Even a Raspberry Pi (a small singleboard computer) can be enough for a system with small demands.

For better evaluation here are three popular operating systems and their requirements:

Operating system	Minimum system requirements
TrueNAS Scale	2 Core x64 Processor, 8 GB RAM, 16 GB Boot device, One drive (two drives very much recommended)
Unraid	1 Ghz 64-bit Processor, 4 GB RAM, Linux hardware support, Two drives recommended, Boot from USB possible
OpenMediaVault	Any processor, 1 GB RAM, One Drive

Please note: For TrueNAS only choose „TrueNAS Scale Community Edition“. The old version „Core“ is obsolete and should not be used these daysv.

If want to make your decision based on software features, take a look at the following table:

Topic	TrueNAS Scale	Unraid	OpenMediaVault
Ease-of-use	Easy to understand, clear structure	Easy to understand, clear structure	User interface limited to basics
Special hardware required?	More RAM leads to better performance (used as chache)	Not really	Minimal requirements
File systems	ZFS	XFS, BTRFS	EXT4, XFS, BTRFS
Drive support	Disks with the same size are highly recommended	Support for drives with different sizes	Support for drives with different sizes
Additional features	App-Store, Docker, Linux Container, VM	App-Store, Docker, VM	Plugins, Docker
Support	Documentation, Community-Support	Documentation, Support from the manufacturer	Documentation, Community-Support
Cost	Free	49$, 109$ or 249$	Free

Here is a video about a very in depth video about the different kind of filesystem:
Linux File Systems Explained for Beginners: Ext4, BTRFS, XFS & ZFS

If you prefer a more practical and written guide, here you go:

BTRFS explained by Synology

ZFS explained by TechTarget

You can and should do your own additional research. Browse the websites of each manufacturer, read the „getting started“ chapters of each documentation, watch some videos on YouTube and read one or two tests. The internet provides many ressources for getting to know NAS softwares.

If you have more experience, you can think about these questions, too:

Question	Your answer
What NAS software do I already know or have seen?	
What security features do I know or want?	
How does the layout and structure of my data looks like?	
What RAID configuration do I want?	
How many backups do I want? Offsite? Cold-storage? Cloud-backup? (A graphical representation is recommended for this)	
Do I want to use a hardware or software RAID controller?	

Read more about RAID controllers here:
[RAID Controllers explained by TechTarget]

Choosing appropriate hardware

If you don't want to choose each pc component individually, you can - and should - read on to get to know the components that are in such a system. Especially the different types of storage devices should be known to you. Also you will find an overview of different ways of acquiring NAS hardware.

If you want to choose individual parts, they have to be compatible. You can check the compatibility in the product description of every part. For example: A central processng unit (CPU) with a "1700 sockel" won't fit in a motherboard that is made for a "1150 sockel". If you want to be on the safe side, you can use a "pc buidl function" from one of the many online shops. These will only display parts to choose from that are compatible with each other.

On the next pages you will find some notes on the key components and what to look out for.

Central Processing Unit (CPU)

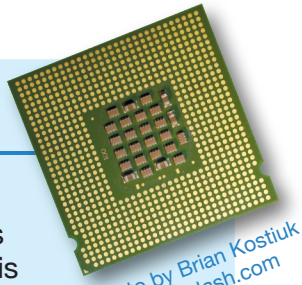


Photo by Brian Kostiuk at Unsplash.com

Most cpu's mention the word "Cores" and "Threads". A core is a compute unit that does calculations (tasks). How fast it can do these calculations is defined by the "GHz" (higher is better). A "thread" is a „core“ split in half. This enables the cpu to do more calculations at the same time.

If you want more performance choose a cpu with more cores, threads and higher "GHz". Please note: This will increase your costs and energy usage.

For basic tasks like file transfers, backups and one or two additional apps a 2-Core (4 Thread) cpu is good enough.

For basic setups low-power processors like an „Intel Pentium“ are sufficient. For more demanding use cases, consider Intel Core i3 / i5 / i7 or AMD Ryzen processors.

A low-TDP (Thermal Design Power) cpu is an indicator for reduced energy consumption but please note that the overall power consumption will also depend on your other parts.

CPU-Cooler

Ensure the cooler is compatible with your chosen cpu and pc case. Check the mounting mechanism and dimensions for that. Also opt for a quiet and efficient cooling solution; air coolers like Noctua or BeQuiet are good options for that (as of 2025).

Only choose a watercooling solution if you are familiar with it. Even if a friend can help you buy and build it in your system he/she may not always be around if any problems accure.

For a NAS, prioritize coolers with minimal upkeep and durable designs. Stock coolers that come with a cpu or those already built in a refurbished system are most often good enough for basic systems but may be louder than more expensive coolers.

Random Access Memory (RAM)

Decide based on your NAS use case: For simple file storage, 4–8GB is sufficient (TrueNAS Scale: more RAM leads to better performance because of caching).
For virtualization or media editing, aim for 16GB or higher.

Match the RAM type to your mainboard specifications (e.g., type „DDR4“ or „DDR5“, speed „2666“ or „3200“, ...).

If you want added reliability, consider RAM with Error-Correcting Code (ECC) support. This type of RAM can handle errors while transferring files on it's own rather that loosing the data.

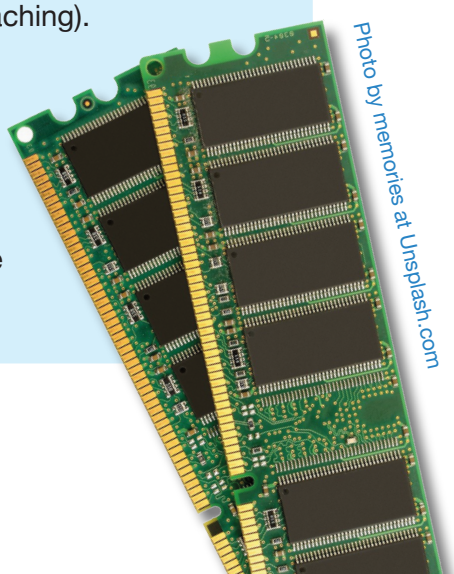


Photo by memories at Unsplash.com

Graphics Processing Unit (GPU)

A graphics card is not needed for a basic and moderate NAS unless you want "to build your own netflix". If so you have many video files to store and stream and therefore a need for something called "video transcoding". Video transcoding is the process of converting a video from one format to another. For example, you might change a video file from one type of file (like .MOV) to another (like .MP4). This can also involve adjusting the video's quality, size, or resolution to make it easier to play on different devices or upload to the internet. Essentially, it's like "repacking" the video into a new form that's more suitable for your needs. Since this is a resource intensive task, a gpu is recommended but many cpu can do video transcoding - to some extend - as well.
Tip: If you are unsure, build a NAS without a gpu but leave room in your case to include one. Then you can test your system and see if you really need one.
But be warned: A gpu will increase your energy consumption (by a lot). Most of the time a gpu will consume equal to or more power than all other parts - together!

Power Supply (PSU)

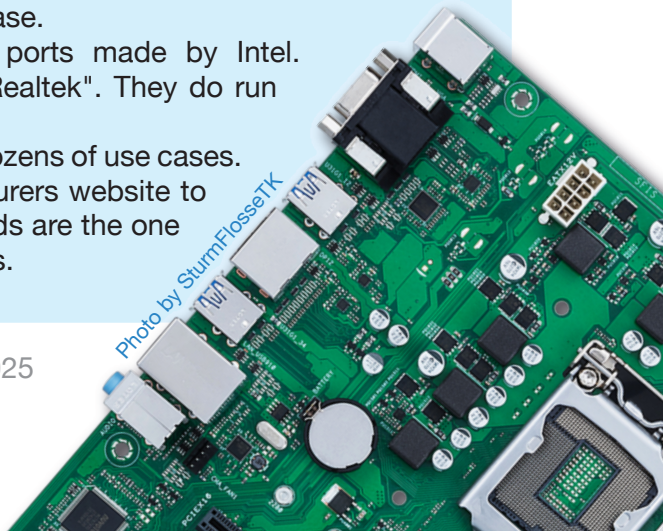
Choose a psu with enough wattage for all components and maybe some headroom for upgrades (usually 300–500W for NAS setups). Look for psu's with an 80+ rating (e.g., bronze, gold, or platinum) for better energy efficiency. A modular psu can be helpful for cable management and investing in reputable brands ensures long-term stability.

Additional fans

Fans help to keep you drives and other components cool which extends their lifespan. You need to choose an approriate fan size for your case (e.g., 120mm or 140mm). Select fans with low noise levels, especially for a NAS that's running 24/7. Good manufacturers are BeQuiet and Noctua (as of 2025).

Mainboard

Ensure the board has adequate ports for your storage drives. Match the mainboard's size to your needs (e.g., mini-ITX, micro-ATX) and pc case.
Look for mainboards with built-in gigabit network ports made by Intel. Most often cheaper boards include ports made by "Realtek". They do run well but have more issues than those from Intel.
There are all sorts of motherboards because their are dozens of use cases. If you found one that seems to fit, open the manufacturers website to check the intended use case. Gaming and server boards are the one that draw more power than those for „simple" pc builds.



PC-Case

Ensure the case has room for your mainboard and all other components, while fitting in the intended location.

Look for sufficient drive bays for your storage devices, plus room for potential expansion.

Choose a case with good airflow and space for additional fans. Consider cases with sound-dampening materials for a quieter NAS. Dust filters keep your build cleaner.

Storage devices

When choosing your storage devices you need to think about three categories: What type of drives do I want / need? Should I buy new, refurbished or can I use my old drives? How many drives should I use?

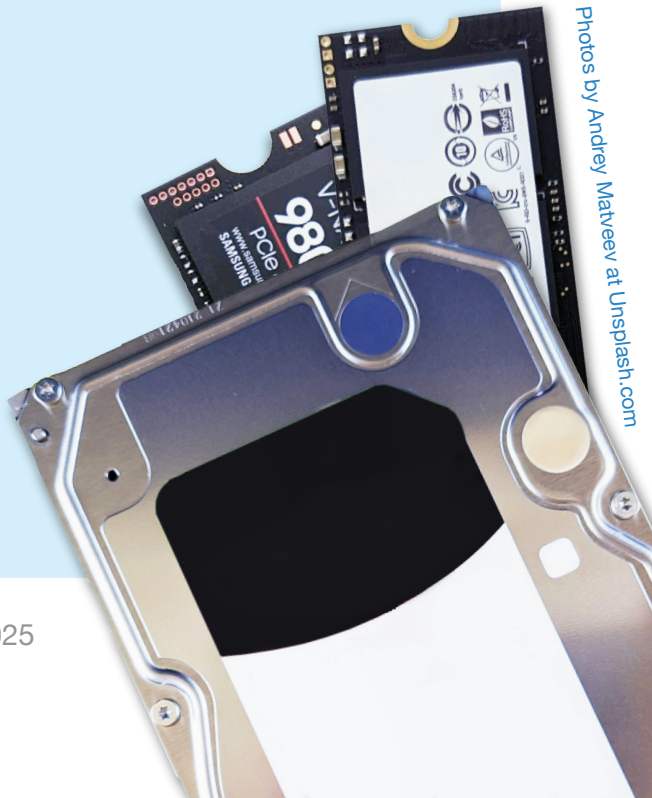
Choosing a type of drive

There are different kinds of storage devices for different use cases:

- **Non-Volatile Memory Express (NVMe)**
    - Main use case: Boot devices
    - Data transfer: Fast
    - Price per TB: High
    - Energy consumption: <1 W
    - Connection: Connector on mainboard required, no cables
    - Build quality: Small, fragile, no moving parts, cooling pad recommended

- **Solid State Drives (SSD)**
    - Main use case: Fast storage or boot device
    - Data transfer: Medium
    - Price per TB: Medium
    - Energy consumption: 2-3 W
    - Connection: SATA for data and power
    - Build quality: Slim, no moving parts

- **Hard Drive Disk (HDD)**
    - Main use case: Big storage
    - Data transfer: Slow
    - Price per TB: Low
    - Energy consumption: 5-10 W
    - Connection: SATA for data and power
    - Build quality: Strong but many moving part



Photos by Andrey Matveev at Unsplash.com

It is recommended to choose drives that are the same size because some NAS operating systems will only use the capacity of the smallest drive as "usable space" for each other drive. This is because the drives will later be put into groups for data referencing (one drive knows what data is on the other drive) and the amount of data on the bigger drive won't fit on the smaller drive (more on that later in the chapter „RAID“).

New vs. old vs. recertified („refurbished“)

A general rule of thumb: Buy and use new hard drives for your main NAS as they will have the longest lifespan and warranty. Old hard drives are only suitable for a secondary NAS (to replicate the main NAS) and for a single purpose backup server.

You should check all drives carefully. For example: Check the physical state of each drive and use a program like "CrystalDisk Info" to view the used lifespan and the amount of startups the drive went through. This way you can compare drives and evaluate if you are comfortable using the drive for your important data.
If you do not feel comfortable using a drive, don't! Money spent on reliable hardware will not keep you awake at night as much a bad drive or failing system!

If you don't have the money to buy new drives but you also don't feel comfortable going with your old used drives, an alternative can be "factory recertified" drives. These are "already been bought" drives that have not seen much use in businesses (e.g., as a spare drive) and are again reclaimed, checked and refurbished by the manufacturer and offered at cheaper prices than new drives. It's unclear what kind of drives you get and you only have a short time of warranty (usually six months) but it's an option, too. **But be careful:** Some shops sell "seller recertified" drives. These are not the same as the manufacturer does far more intense testing than a seller ever could!

Generally speaking you should also buy from manufacturers with a good reputation. For this you can read tests and reviews online. Regarding big storage hdd’s in 2025 you basically have the option to buy from two main hard drive manufacturers: Seagate or Western Digital (WD). You can't go wrong with either of these but choose drives that are built for continues use - like Seagate Ironwolf or Western Digital Red.

How many drives do I need?

When buying drives for your NAS you need to buy more drives than you think you need because in most cases you want hardware redundancy. This means that you protect your data from hardware failures and you don't loose any data if on drive fails. You are building a "Redundant Array of Independent Disks (RAID)". We will dive into this topic more clearly in a later chapter but for now know that you need to choose a level of hardware security for your data before you buy your drives. You can read on before buying anything, of course!

Also as the amount of data usually gets more rather than less it is recommended to plan for the long term. A good plan now is cheaper than a second system in the future.

Here are two videos about expanding storage:
TrueNAS Tutorial: Expanding Your ZFS RAIDz VDEV with a Single Drive

Fixing Linus' Home UNRAID Server (Replacing + Expansion)

Here is a short list of RAID configurations and their individual hardware tolerance:

- **RAID 0 (Stripe) - not recommended for storing data permantly!**
    - Storage capacity:
        - Full
        - Every GB on every drive can be used
    - Hardware tolerance:
        - Zero
        - **One drive fails means all data is gone**

- **RAID 1 (Mirror)**
    - Storage capacity:
        - One drive
        - Data on drives get duplicated. Multiple drives only include copies of data
    - Hardware tolerance:
        - With 2 drives, one can fail before loosing any data
        - With 3 drives, two can fail before loosing any data
        - …

- **RAID 5 (Single „parity“)**
    - Storage capacity:
        - (N-1)/N
        - Uses one drive's worth of space for „parity“ (a special storage location)
    - Hardware tolerance:
        - Moderate
        - Can tolerate the failure of one drive; data can be reconstructed using parity information
    - Benefit against RAID 1: Expansion with one instead of multiple drives

- **RAID 6 (Double „parity“)**
    - Storage capacity:
        - (N-2)/N
        - Uses two drives for „parity“ (providing extra redundancy)
    - Hardware tolerance:
        - High
        - Can tolerate the failure of up to two drives without data loss
    - Benefit against RAID 1: Expansion with one instead of multiple drives

Please know that there is also a RAID10 (a combination of RAID 0 and 1) but since this explanation is for beginners it is not mentioned here. If you want to know more about it, you can read the chapter about all RAID configurations.

A visual representation of all RAID-levels is also displayed in the chapter about „RAID“.

Example - You want 8 TB of usable space in your NAS. These are your options:

1. You could get a single 8 TB drive and create a RAID 0 („stripe“). This would be the cheapest but also the least safe option because there is no hardware redundancy. If the single drives fails, your data is lost.

2. You can get two 8 TB drives and build a RAID 1 („mirror“). The second drive does not provide usable space but mirrors the first drive and provides the data if the first drive fails. You can buy a third drive and include it into the mirror to get more redundancy but not more usable space (expansion of usable space is not possible with a single drive after building the mirror with the first two drives).

3. You can get two 8 TB drives and build a RAID 5. In this case one drive is used for hardware redundancy ("parity") and the other one is usable space. The advantage of this against RAID 1 is expansion: You can add drives afterwards to expand your usable space. The parity storage stays the same. In this case it would only be beneficial to buy another 8 TB drive, not a smaller drive.

4. You can get four 4 TB drives and build a RAID 6. Two drives provide hardware redundancy („parity“) while the other two drives provide usable space. Afterwards you can expand your storage by adding another 4 TB drive at smaller costs than in RAID 5 but also with smaller overall storage.

The more robust the level of redundancy, the more drives you need, ultimately reducing the amount of usable storage relative to the number of drives you have. Always account for this overhead when planning your NAS setup.
If you want to do it perfectly, think about the costs of replacing drives, too. In the example above buying another 4 TB drive is cheaper than an 8 TB drive in the future.

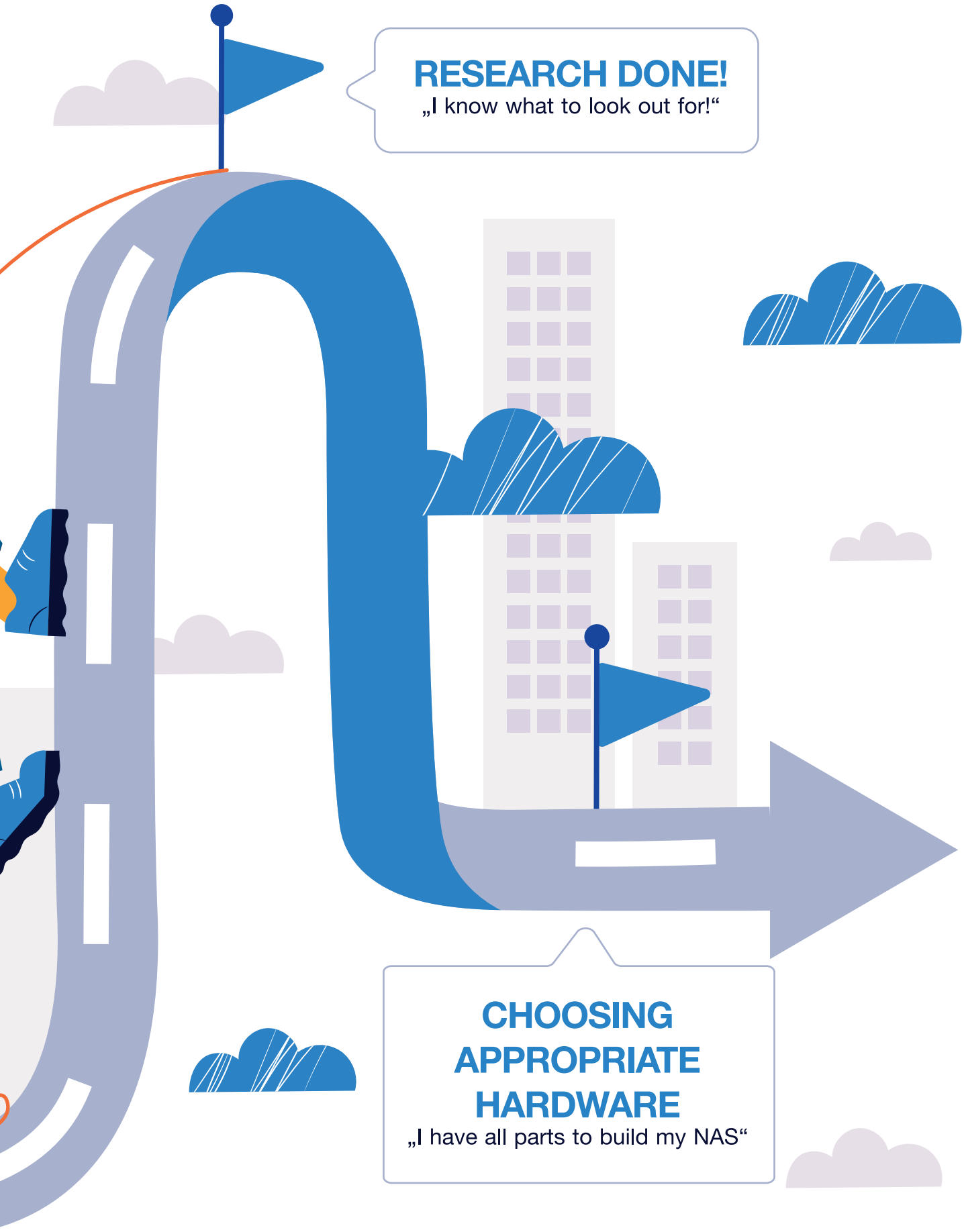
If this is all overwhelming for you, dont’t worry. You can take a look at the case study at the end of the manual or get inspiration by other users on the internet.

Keep in mind:
You need **„RAID configuration +1“** drives for your system because you need a place to install the operating system.

You are building a system where individual components need to be compatible with each other. This takes some time… and that is normal… and ok.

If you can’t decide between two options, speak to a person that knows more about this than yourself, post your specific question in one of the many forums or reddit to get feedback from the community, watch one or two youtube videos or do additional research on the internet. You will find your way!

PERSONAL CRITERIAS
„I want to build my NAS!“
START
„I want to build a NAS“



Designed by pikisuperstar at Freepik.com

Designed by Freepik at Freepik.com

Assembling the hardware

Ok, theory out of the way. Time to get our hands dirty!

If you already built a custom pc before, you can go straight ahead. Put the components together like you used to. Just take "future plans" and "expandability" in mind as well. You will thank yourself later if you do a clean build which is easy to clean and nice to work in - for example if you need to replace a faulty drive or switch out components to get more performance.

Have fun!

If you need help please take a look at the note section at the end of the next page.

Read the next chapter if you are finished!

If you never built a PC before, you still have a bunch of options:

1. **Buy a prebuild NAS from Synology, Qnap or Ugreen**
   - This will give you a complete system (with or without hard drives)
   - If you choose such a system, please read again the disclaimer at the beginning of this manual
2. **Buying a used NAS from someone else**
   - This option is not recommended because you don't really know why the person is selling the NAS
   - Future failures of the system will cost you more (e.g., the loss of data) than money
3. **Buying a new pc**
   - This will give you the most ideal situation but will also cost you the most
   - It requires some pc knowledge to select parts that fit together well but some shops have tools on their website that check compatibility (e.g. Alternate)
   - Please note: In most cases a NAS does not need to be that powerful
4. **Buying an old but refurbished office pc**
   - This is a suitable option if you don't want to choose and buy new hardware and when you don't already have pc parts on hand
   - A refurbished office pc contains hardware that was used in business but is not powerful enough anymore for their use case and was checked by professionals
   - For a NAS most refurbished pc's will be powerful enough
   - Please note: Many old office pc's have propriotary hardware. This means you can't easily replace parts if they fail. To find out which builds include proprietary hardware you can compare pc's and ask yourself "Could I put this part into another PC?" or read reviews and do additional research.
5. **Use your old spare computer that you already have on hand**
   - Using this you will save money and time but you need to check your parts carefully, if they are compatible, applicable and still safe to use (e.g. Power supply)

Building a custom pc is not difficult if you understand what each part of the system is there for and where it needs to be. You can take a look at the case study at the end of the manual or follow these great guides on YouTube (click the link or scan the qr code):

How to Build a PC, the last guide you'll ever need! (2024 Update)

If you prefer a more „person of view“ style of guide:

First Person View POV PC BUILD Guide! (2024)

Installing the operating system

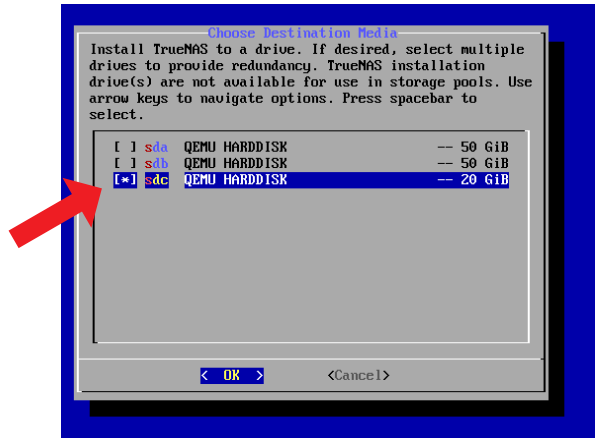
For installing the operating system you need:

- Your finished NAS build
- A display connected to the NAS
- Mouse and keyboard connected to the NAS
- A USB Stick (16GB or bigger)
- A network connection connected to your NAS (preferable a cable connection)

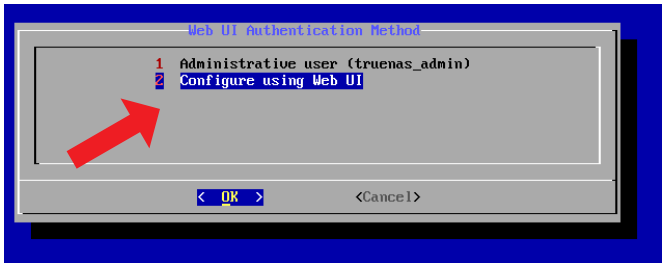
Here is a step-by-step guide on how to install the operating systems:

1.) Download the ISO from the manufacturers website.
2.) Create a bootable USB using tools like Rufus or balenaEtcher.
3.) Plug the USB into your NAS and enter the BIOS (usually by pressing del, F2, F12 during startup).
3.1) If you wanna create virtual machines, you need to enable VT-d and VT-x.
4.) Got to „Boot options“ and set the USB drive as the primary boot device or boot directly from the usb.
5.) Follow the on-screen installer prompts (shown in TrueNAS Scale):

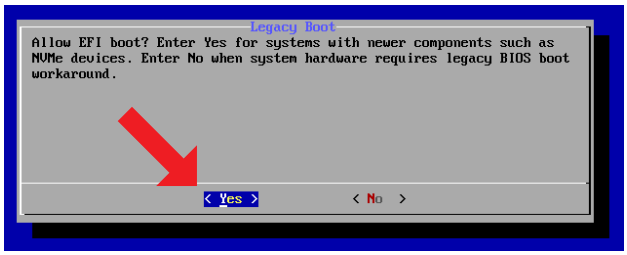
In the installation process you will be asked to select the drive(s) you want to install the operating system to. You can select the drive(s) with the space bar (a star will appear next to the selected drive(s)) and confirm your selection with „Enter“:


Choose Destination Media
Install TrueNAS to a drive. If desired, select multiple drives to provide redundancy. TrueNAS installation drive(s) are not available for use in storage pools. Use arrow keys to navigate options. Press spacebar to select.
[ ] sda QEMU HARDDISK -- 50 GiB
[ ] sdb QEMU HARDDISK -- 50 GiB
[*] sdc QEMU HARDDISK -- 20 GiB
< OK >
<Cancel>

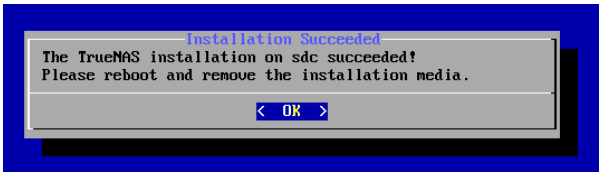
You will also be asked how you want to configure your first admin account. It is recommended to select „Configure using Web UI“. Then you can select your password when logging in the webui for the first time.



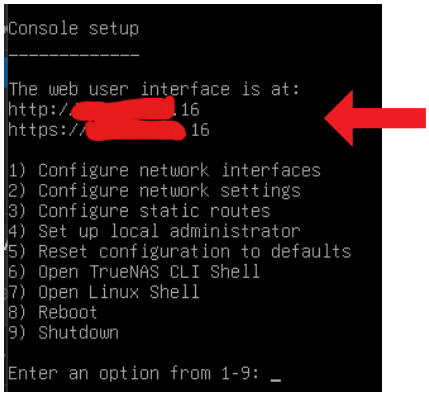
When you can choose to enable „EFI boot“, select „Yes“ if your hardware is not older than ten years (for example when using NVMe drives).


Legacy Boot
Allow EFI boot? Enter Yes for systems with newer components such as NVMe devices. Enter No when system hardware requires legacy BIOS boot workaround.
< Yes >
< No >

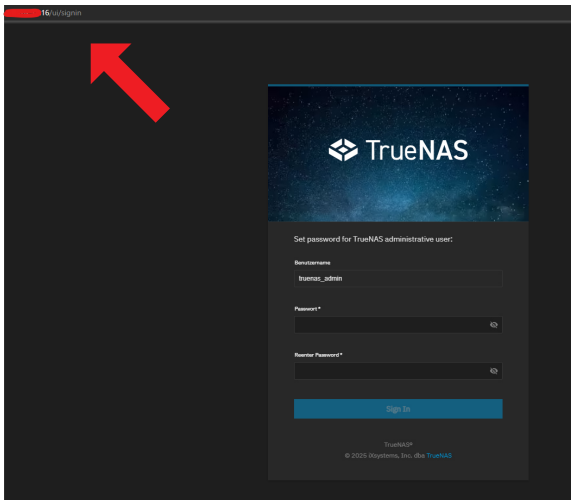
After the installation progress you should see this:



6.) Once installed, shutdown the system, remove the USB, startup the system again.
7.) Check the ip address shown on the display that is attached to the NAS after startup:



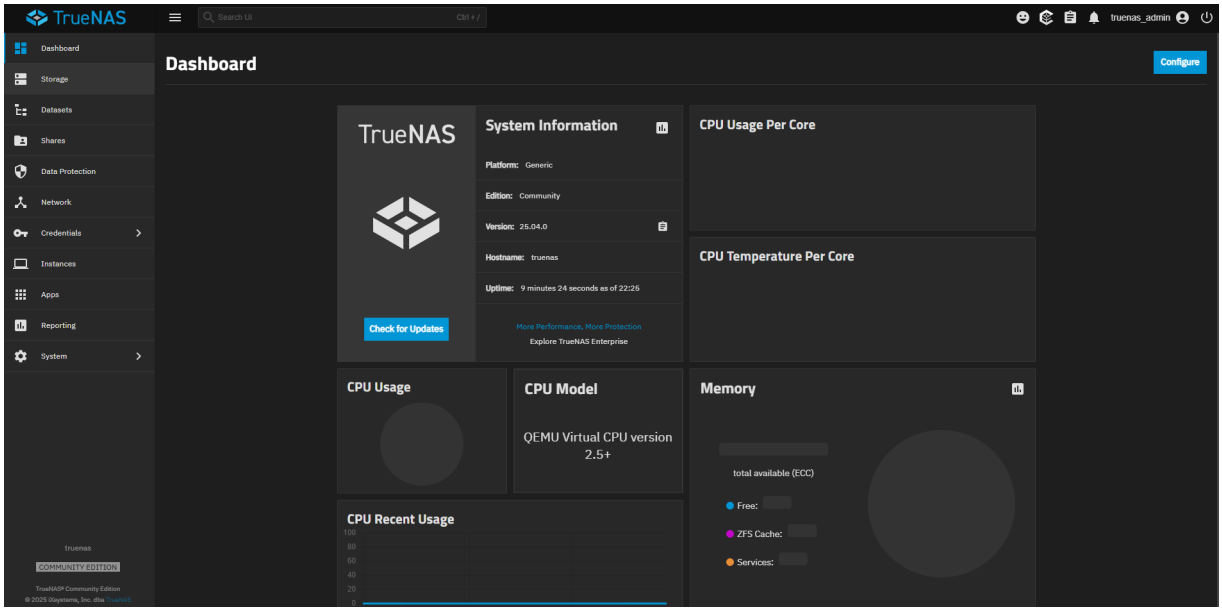
8.) Access the webui from another device by entering the ip address into your browser.



There you can enter the password of your choice. Choose a strong and secure password since this will be an admin account who will have access to all settings!
If you can’t think of a secure password you can create one using this popular website: Bitwarden Password Generator

Note down your password and store it in a safe place!

After logging in you should see the dashboard:



This is the place where you can configure your whole system. It is available on every device that is connected to the network the NAS is also connected to but you will always need to enter the name of the admin and the password to configure something on your NAS.

After confirming that you can successfully log into the webui you can remove the display, mouse and keyboard from your NAS hardware and put the NAS into it's final location with a network connection attached. **Just remember to shutdown the system via the webui rather than just pulling the cables before moving the hardware!**

Software concepts

You have turned an idea with personal criterias into a concept and transformed this concept into a real hardware setup. **Well done!** You are half way through!

Now we need to know how to configure the software to control the hardware and tell each component what it needs to do.

A NAS operating system has a special use case and is therefore not like any normal operating systems (like Windows, Mac or Linux). It contains special features and configurations that you need to know about.
Don't panic! Most of these are simple once you get the hang of it. The following chapters will explain the most important concepts in a simple and easy-to-understand language.

Please note:
The software concepts will use the words of the operating system "TrueNAS Scale" but the concepts will be roughly the same in every operating system. You can follow guides in the documentation of each individual operating system or on YouTube to find the appropriate buttons in each individual graphical user interface (GUI).

Virtual devices and RAID

Implementing drives in a diy NAS is a little different than using a single external drive attached to a pc - as you will know after reading the part about storage devices.
In our case we want to access one big storage from any user on the network that has access rights. Of course we could find a single drive which has the exact size that we need and use only this. But even if this drive is bigger than your current needs that scenario would work against one of our core concepts called "availability". What if this single drive fails? Even if you could restore your data from backups is it really suitable to have downtime (no access to the data) each time a single drive fails? In most cases it is not.

For being able to access our data even when a single drive fails we need to combine multiple smaller drives into one big storage. In the NAS-World this is called "Creating a vdev" - a virtual device - or "Redundant Array of Independent Disks (RAID)".

There a different ways you can build a RAID but two things are common in most NAS operating systems:

1.) The storage capacity is limited by the smallest drive, so an 8TB and a 4TB drive for example would be similar of using two 4TB drives because by combining multiple drives each one "knows" what data is stored on the other drive. This way we can rebuilt our storage if one drive fails. But a small drive can not reference everything that is on the big drive. The extra 4 TB on the 8 TB drive therefore would not be utilized. This is why mixing drives of different sizes is not recommended. Some operating systems can still use different drives.

2.) A RAID can be created by multiple vdev's. For example we can combine four drives into pairs of two. With those two pairs we create another RAID. But we have to keep one thing in mind: If one pair (one vdev) fails all data is lost.

In the following you will find a breakdown of the most common methods of building a "RAID". Regardless of that…

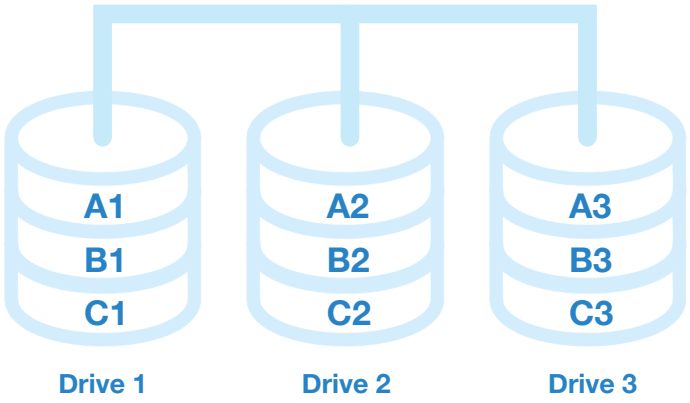
Please remember:
RAID is not a backup and therefore does not protect you against file deletion or corruption!
RAID is only for hardware redundancy!

RAID 0 (Stripe)

RAID 0, often called striping, is a configuration designed for maximum performance, requiring at least two drives. It works by splitting - or “striping” - data across all drives in the array. This allows multiple drives to work simultaneously, significantly improving read and write speeds compared to using a single drive. One drive can provide „A“ while the other provides „B“ - at full drive speed.

Please note: If a single drive fails, all data is lost.

WARNING: You read right. If a SINGLE drive fails (regardless of how many are in the array), ALL data is lost! Their is NO hardware redundancy! Stripes multiply risks of loosing data you rely on! They should only be used for special use cases (temporary storage, cache, etc.)!

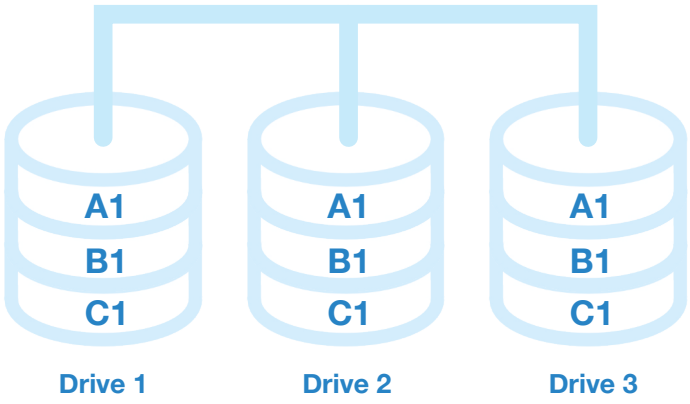


RAID 1 (Mirror)

A mirror contains out of at least two drives where each drive contains exactly the same data. This is the safest way to configure a virtual device because you have an exact copy of all data on another drive. When a drive fails, replacing it copies the whole data from the still functioning disk onto the new drive to "rebuild" the mirror.

The disadvantage is capacity: If you use another whole drive as a copy the only free space to fill is a single drive. With two drives in a mirror you can only use 50%, with three drives just 33% and so on.

Please note: Data will be lost, if all drives fail.

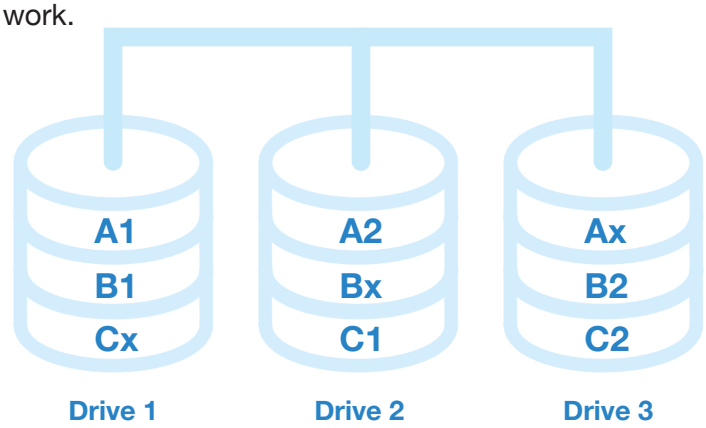


RAID 5 (RAID Z1)

A RAID 5 contains of at least two drives where each drive contains parts of the whole data stored in the array. The size of one drive will be used for "parity", a special type of storage that is stored partially on each individual drive. The information stored in "parity" reflects information about the data that is stored on all the drives. With "parity information" we can rebuild the data on a failed drive using the information stored on the other drives.

Regarding storage you will loose the size of one drive - no matter how much drives are in the array.

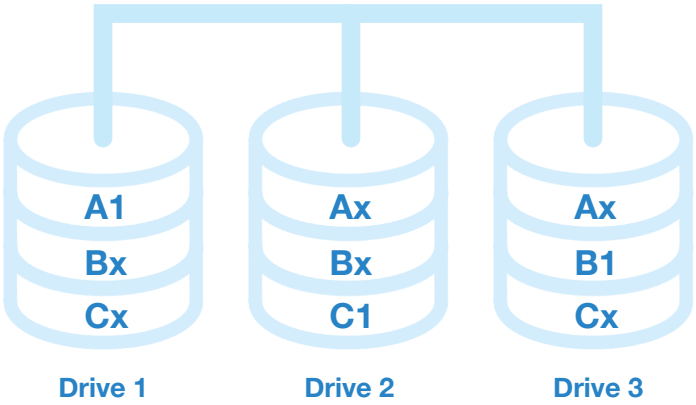
Please note: If more than one drive fails, all data is lost. There is no way of rebuilding the data even if the other drives still work.



RAID 6 (RAID Z2)

RAID 6 is basically like RAID 5 but uses the size of two drives for "parity" instead of one. So you need to calculate your storage needs with the addition of two drives for "parity". But you also have more redundancy because up to two drives can fail before you loose any data.

Please note: If more than two drives fail, you will lose all data in the array.



RAID 10 (RAID 1 + 0)

This configuration combines the principles of RAID 1 and RAID 0, offering both redundancy and performance. It requires at least four drives, with data being both mirrored and striped.

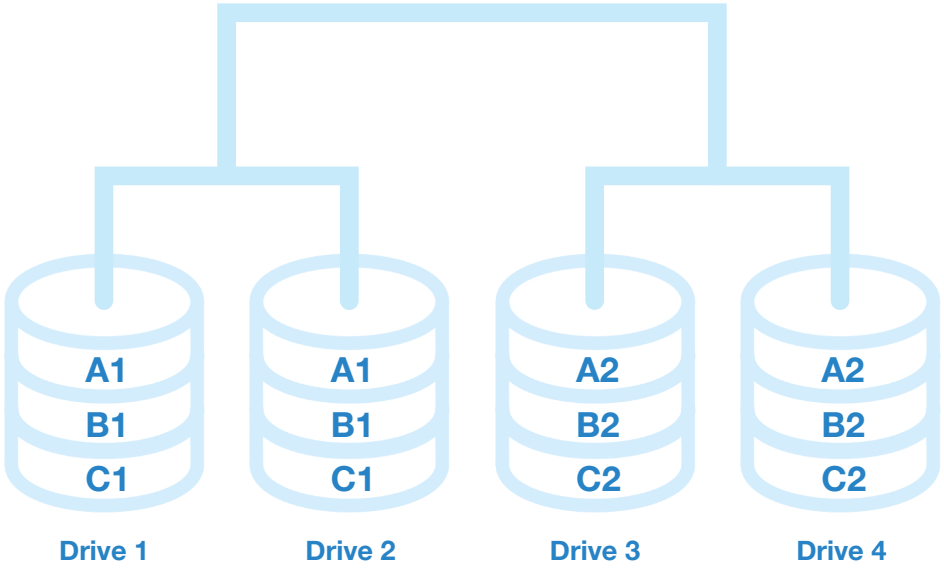
Here's how it works: the data is mirrored across pairs of drives (just like RAID 1), ensuring that if one drive fails, its identical copy on the mirrored drive remains intact. At the same time, the mirrored data is striped across multiple pairs of drives (like RAID 0), enhancing read and write speeds by distributing data access across multiple disks.

Regarding storage, you effectively lose half the total capacity of the drives in the array, as one drive in each mirrored pair is used solely for redundancy. This means, if you have four drives of the same size, you'll only have the combined storage capacity of two drives available for data.

RAID 10 offers excellent performance and high fault tolerance, but it comes at the cost of requiring more drives and sacrificing half of the usable storage for redundancy.

Advantage: It’s often a choice for systems where both speed and data protection are equally critical.

Disadvantage against RAID 6 (Z2): If one of the vdev's (pairs of drives) fails all data is lost because overall it is a „stripe of mirrored vdevs“. And if a vdev in a stripe fails, all data is lost. So it can happen that two drives in RAID 10 fail and all data is lost but two drives in RAID 6 fail and all data is still intact. If you want more data protection instead of speed, choose RAID 6.



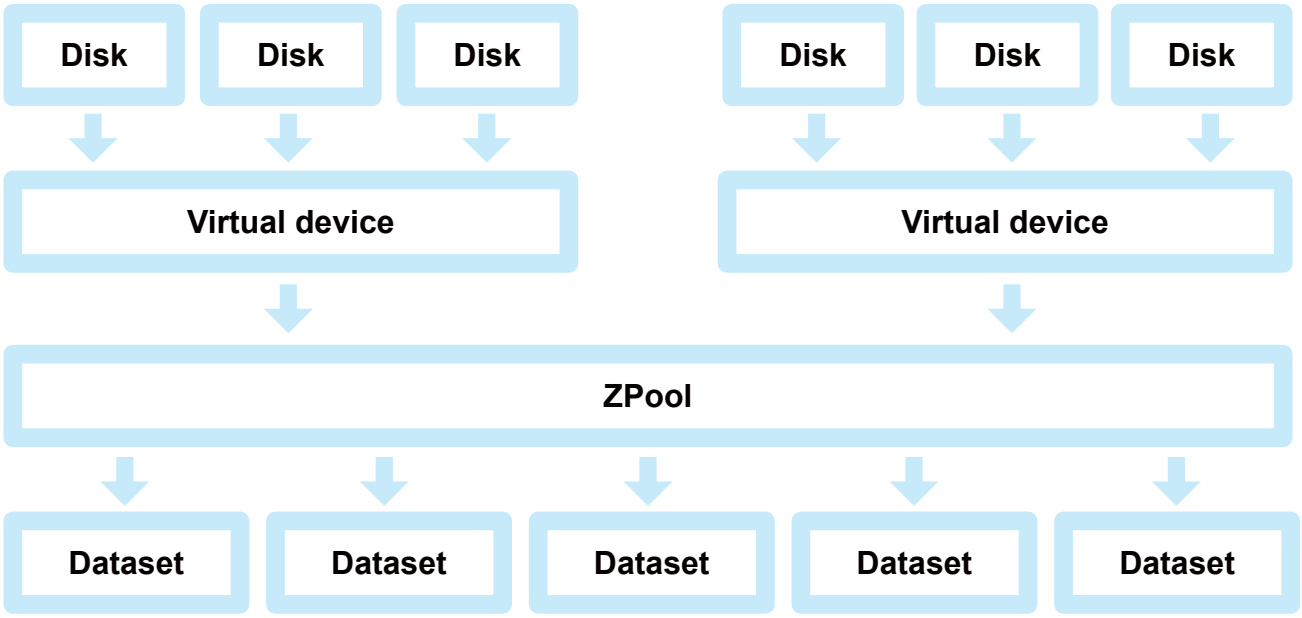
Again, RAIDS can be expanded but you need to plan your expansion well:

TrueNAS Tutorial: Expanding Your ZFS RAIDz VDEV with a Single Drive

Fixing Linus' Home UNRAID Server (Replacing + Expansion)

Graphical summary

In total you are using drives to create "virtual devices". With "virtual devices" you create a "data pool" or "zpool" ("z" stands for "ZFS"), which is as big as the virtual devices in their RAID configuration. In the "data pool" you create "datasets", a set or group of data.



Note on pool creation and encryption

When creating a pool you have the option to encrypt the pool. Turning encryption on is recommended.

This way a special key is created that is stroed in the configuration file of the system to unlock the data in the pool. This means that if the drive get stolen and inserted into a system that does not have the key, the data is still unaccessible and safe. Your system does know the key and will unlock the pool on each restart automatically for you.

The disadvantage to that is that if your whole system gets stolen the config will unlock the pool and the data would be accessible if the thiefs know your user passwords or a way to access the drive another way.

To prevent this issue you can change the encryption type to "passphrase" and define your own password. **The „passphrase“ does not get stored into the configuration of the system** and therefore drives will not automatically unlocked. **You need to note the passphrase yourself in a safe location!**

The downside is that you will need to login the webui each time the system gets restarted and unlock the pool and datasets manually.

You can define key or passphrase encryption for each dataset individually.

Recommendation: Set key encryption (automatic unlock) for datasets that contain services (apps) and non-sensitive data. Set passphrase encryption for important and sensitive data.

Datasets

Datasets can be viewed as a group of data or a smaller part of the overall storage which has it's own settings regarding permissions, access rights, sharing options, backup settings and more. It is possible to create child-datasets - a dataset within a dataset.

Here is an example of a simple dataset layout:

- Users
  - User 1
  - User 2
- Media
  - Photos
  - Videos
  - Music
  - Books
  - Documents

Each user could have his/her own dataset with their own data and the permissions are set so only the users can access their own dataset. Every dataset of every user can be a child-dataset and be backed up by one recursive backup task of the dataset "users".

Then each user has access to the media dataset and every child-dataset in it. Once again one recursive task can backup each child-dataset in "media".

Dataset layout

Carefully planning your dataset layout is highly recommended because this way you will have a clear understanding of where data will be located and where it should be stored by the users. If you create dataset after dataset with huge amounts of data already in them you will end up copying data multiple times and eventually have a huge mess. A clear structure acts like a guide for you and other users.

Recommendation: Create a visual representation of your layout with notes on the type of data and backup rules you want to implement. Always think about: Does this data fit together in one group? What data is more important than other (and should therefore backed up more frequently)?

Dataset permissions

Which permission or access right options you can configure depends on the dataset preset you choose. Permissions are also called "ACLs" (Access Control Lists).

Here is a short explanation of the permission differences in TrueNAS Scale:

- **Generic:**
  - Permissions: Granting the owner full control, while the group and other users have read and execute privileges
  - Use Case: Ideal for non-SMB share datasets or datasets for virtual machines and containers

- **SMB:**
  - Permissions: Inherits ACL permissions from the parent dataset. If no ACL exists, permissions are calculated to grant full control to the owner, group, and administrators, with modify control for users
  - Use Case: Optimized for SMB shares

- **Apps:**
  - Permissions: Similar to the SMB preset but includes additional entries granting modify control to application-specific groups
  - Use Case: Designed for application storage, ensuring smooth operation for apps requiring specific access rights

- **Multiprotocol:**
  - Permissions: Inherits ACL permissions like the SMB preset, but optimized for both SMB and NFS multi-mode shares
  - Use Case: Suitable for environments requiring both SMB and NFS access

Most often it is recommended to use the "SMB Preset" where you can individually choose if you actually want to create an actual share to access the data on other machines or if you just want the additional access rights that come with this preset.

SMB share

Each dataset you create will not be accessible to any user on the network without a SMB share. SMB stands for "server message block" and is a network protocol used to access data and information of other systems over the network.

In most operating systems it is called a "service" that can be started, restarted and turned off. If you add a new SMB share you need to restart the service to implement the change.

Backups

Backups can be done in many ways. The standard and best implemented solutions in the operating systems are snapshots, rsync and zfs-replication but you can also install additional software on the NAS to get more fine-grained control if you need to.

A well-thought-out dataset layout enhances precision during regular backups, as most systems are designed to backup entire datasets using snapshots rather than selecting individual files or folders. A good layout enables you to backup some datasets more frequently than others and store some backups longer or shorter than others.

A recommendation for creating easy-to-backup datasets is to create smaller and more precisely defined rather than larger undefined datasets.

For example: If you have a single dataset that is accessed by multiple users and you want to restore data from one user, restoring the complete dataset to a previous state would affect files from other users as well. If each user has their own dataset you can restore datasets without affecting the others. Also you can implement different rules for regular backups.

Tip for well organized backup-jobs:
You can create one backup job which is backing multiple child-datasets. For that you can use the "recursive" option to backup the selected dataset itself and all child-datasets. All datasets will then be backed up individually and can be restored individually but have the same backup frequency.

Full vs. Incremental backups

Generally speaking there are two different types of backups: incremental and full backups.

Incremental backups save only the changes made since the last backup (whether full or incremental). This approach is faster and uses less storage compared to full backups, but requires previous backups to restore the complete data. It's ideal for maintaining up-to-date backups space-efficiently.

Full backups store a complete copy of the data each time the backup is done. Multiple backups will lead to multiple copies of the same data which requires more storage. The advantage to this approach is that every backup can be restored regardless of wether a different backup is accessible or not.

Rsync

The easiest to understand backup option is "Rsync". It is a tool for synchronizing files and directories between systems. It transfers only the differences between the source and the destination (= incremental backup) which minimizes network traffic and the time required.

The disadvantage is that you will only have one copy of your current data - no backups of previous versions of files. If a file does change, the old version of the file gets overwritten.

Use case: Space efficient backups on other non zfs-based operating systems where only a copy of the currently used data is needed.

Snapshots

Snapshots are - as the name suggests - point-in-time copies of data.

Imagine you have a room that you frequently rearrange. Sometimes you move furniture, swap out decorations, or add and remove items. Now, every so often, you take a photo of the room to capture how it looks at that exact moment. These photos are your "snapshots."

Even if you completely rearrange or remove items from the room after taking a photo, the photo (snapshot) remains unchanged and preserves that moment in time. If you ever want to go back to how the room looked before, you can refer to a specific photo and use it to recreate that setup.

In the same way, file system snapshots capture the state of your data at a particular moment. Even if the data changes later (like rearranging the room), the snapshot keeps the old version intact, allowing you to "restore" or view it as it was when the snapshot was taken. This is incredibly useful for recovering lost or corrupted data without affecting your current setup.

Snapshots are read-only (you can't add or delete individual files after the snapshot is taken) and allow quick recovery of previous states by restoring whole datasets, functions like "previous version" by the OS or accessing folders and files at the storage location of the individual snapshot.

Important note:
A file is only recoverable if a snapshot was taken before the file gets deleted. The same way you can only remember how the room looked if you took a photo of it. If no snapshot was taken before the file got deleted, the file is not recoverable!

How much storage will a snapshot consume?

Snapshots in TrueNAS initially consume no to very little space because they simply record the state of the file system at the time the snapshot is taken. However, as data changes or gets deleted after the snapshot, the snapshot cosumes space to preserve the original data. This means that if no changes occur after a snapshot is taken, it will take up virtually no space. If changes are made — such as modifying or deleting files — the snapshot will retain the original versions of those files, effectively consuming space equal to the size of the changes.
For example, if you take a photo of a room and don't add or remove things afterwards you don't have to store anything extra. If you change something and take new photos you have to store the old photos somewhere until you throw them away.
Regarding data, if you for example delete 1.5 GB of data after taking a snapshot, that snapshot will consume approximately 1.5 GB of space to preserve the deleted data.

Where do snapshots get stored?

A snapshot task saves the snapshot on the same dataset that contains the data because it is a reference of where data is stored. It cannot reference data on a different dataset. A snapshot task cannot send the snapshot to a different drive or location.

Use Case:
- Keeping incremental backups for quick restore of whole datasets, files and directories
- Building the foundation of zfs-replication
- Keeping a copy of active and "deleted" data

Tip:
Snapshot task give you the option to enable/disable "taking empty snapshots". Empty snapshots are snapshots with no recognized change of data.

Not storing empty snapshots is usefull for having a better overview of your snapshots (only snapshots with a change will be shown). This way you will find older versions of files more quickly because you are not browsing snapshots with no change of data. Storing empty snapshots is required for restoring complete datasets.

Recommendation:
- Disable empty snapshots for "temporary" or "short timeframe" snapshots (like every 20 min)
- Enable empty snapshots for bigger restore points (like daily snapshots and longer) that you also want to replicate to other systems

Do snapshots store data indefinitely?

Snapshots can be stored indefinitely but to make better use of limited storage space snapshots should contain a deletion date for old snapshots.

This way you can define a time-based rule for "how long do I want to be able to recover data". You can define rules for every individual snapshot task you enable. Hourly snapshot can stay on the system for a shorter amount of time than for example daily snapshots. The automatic (or manual) deletion of a snapshot will release claimed storage space and make it usable again.

Unfortunately there is no option to "automatic delete snapshots when storage gets full" (like when using time machine backups) because then there is no clear rule for "how long am I able to recover a file".

Snapshots will recognize the change of data related to the previous snapshot of the same snapshot task. So a daily snapshot will recognize the change to the previous daily snapshot even if multiple hourly snapshots were taken between the two.

This is why tiered snapshots are usefull. Here is an example:

Snapshot Task	**Snapshot retention**
When does a snapshot gets taken?	*How long is the data recoverable?*
Every 20 and 40 min past the hour	2 hours
Every hour	2 days
Every day	2 weeks
Every week	2 months

With this schedule you achieve the following:

- Everything stored for at least 20 min is recognized by at least one snapshot and therefore recoverable (for at least two hours)
  - Everything stored for less than 20 min is not recoverable
- Data that is on the system for a longer period of time has a later deletion date
- Changed data that is recognized by multiple snapshots - which would take up redundant storage space - gets automatically deleted
- Deletion of snapshots happen within a reasonable timeframe:
  - Everything deleted within less than an hour is recoverable for two hours
  - Everything deleted within less than a day is recoverable for two days
  - Everything deleted within less than a week is recoverable for two weeks
  - Everything deleted within less than a month is recoverable for two months

Replication

A feature of the zfs-filesystem - that is used in TrueNAS Scale - is zfs-replication. It transfers and syncs already taken snapshots (backups) between two or more zfs-based systems. **ZFS replication relies on snapshots. Without a snapshot, no replication!**

This minimizes the risk of loosing backups (old snapshots) in case of hardware failures and enables you to do offsite backups which are backups stored in a remote location to minimize the impact of the four elements and theft on the availability of the data.

Doing zfs-replications requires an ssh connection to the other system. To do replications to a system in a remote location it is recommended to use a vpn connection in addition to the ssh connection.

Use Case: Protecting yourself against hardware failures and other disasters by moving all active and deleted data (snapshots) to another system - in a remote location.

Replication

Here is a quick comparison between rsync, snapshots and replication for better understanding:

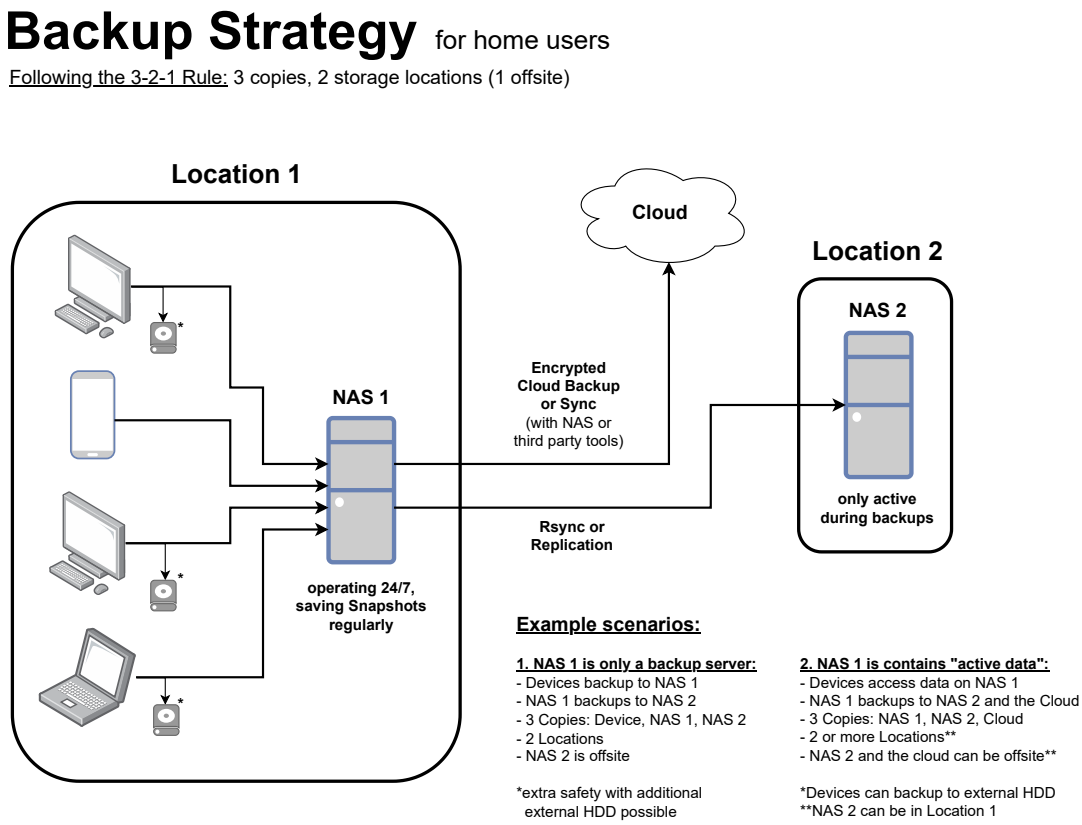
Topic	Rsync	Snapshots	Replication
Purpose	Synchronisation of current files/ directories between systems	Capturing a point-in-time-state of a dataset	Transferring a snapshot to another systems
Storage needs	Requires the same amount of storage as the backup target	Requires only the amount of storage that has changed	Requires the same amount of storage as the snapshot
Data transfer	Works with many types of connections and systems	No data transfer	Requires zfs-compatible systems
Backup location	Different hardware, remote, offsite	Locally, same hardware as data	Different hardware, remote, offsite
Disadvantage	Only the current data is synced to another location	System-specific, local storage	Requires zfs on both the source and destination systems
Use case	Only a current copy is needed, the backup destination does not support ZFS	Fast, incremental backups for easy versioning and rollbacks	Storing snapshots for disaster recoveries and extra safety

The main difference between Rsync and Snapshots is

- Rsync saves only the current data*
- Snapshots saves the current data and also previous versions and deleted files

*with special configuration it is possible to store old files with rsync, too.

Visual representation of a complete backup strategy


Backup Strategy for home users
Following the 3-2-1 Rule: 3 copies, 2 storage locations (1 offsite)
Location 1
NAS 1
operating 24/7, saving Snapshots regularly
Cloud
Encrypted Cloud Backup or Sync (with NAS or third party tools)
Rsync or Replication
Location 2
NAS 2
only active during backups
Example scenarios:
1. NAS 1 is only a backup server:
- Devices backup to NAS 1
- NAS 1 backups to NAS 2
- 3 Copies: Device, NAS 1, NAS 2
- 2 Locations
- NAS 2 is offsite
*extra safety with additional external HDD possible
2. NAS 1 is contains "active data":
- Devices access data on NAS 1
- NAS 1 backups to NAS 2 and the Cloud
- 3 Copies: NAS 1, NAS 2, Cloud
- 2 or more Locations**
- NAS 2 and the cloud can be offsite**
*Devices can backup to external HDD
**NAS 2 can be in Location 1

Cloud backups are not really a topic of this manual because we are searching for data freedom which we don't achieve by relying on the cloud (storage of someone else) again.

Could I not just use the recycle bin (like in windows)?

The answer to that is... yes, you could. That feature is also accessible in systems like TrueNAS Scale but has it's own limitations and is not recommended (= „legacy feature" as of 2025).

When using a recycle bin you are not actually keeping a backup of your data. You are just protecting yourself from your own mistakes (accidently deleting files). You are not keeping a backup of your non-deleted files.

You could combine features like the recycle bin and snapshots to give yourself the option for a quick restore of accidently deleted files (recylce bin) and a backup of all your current data (snapshots) but this will limit your storage space and ease-of-use dramatically. Here is why:

Deleting files to the recycle bin still accounts as "active data" for a snapshot task because the file was just "moved" to a different location (the recycle bin) and not actually deleted. The snapshot does not account this action as a "change of data". The recycle bin will claim the storage space for the files it contains.

If you want to use the claimed storage space from the recycle bin by deleting the files in it, the storage space is not actually usable because the snapshot will then recognize the files removed from the recycle bin as "deleted data" and claim the storage space to enable you to restore this data (that's what a snapshot is there for). So it is basically a "double deletion": moving files to the recycle bin and then deleting all snapshots that hold on to this data to reclaim storage space. Then by deleting old snapshots you are preventing yourself from restoring other data of this snapshot.

The solution to this is: Define or re-adjust snapshot schedules and retention rates to your needs instead of using the recycle bin!

The man difference between the "Recycle bin"-feature and Snapshots are

- **Recycle bin**
  - enables you to move files to a special location
  - protects you against accidental deletions
  - makes a file recoverable immediately (without the need to run a snapshot task)
  - "deleted" data still accounts as active data for a snapshot task
  - "deleted" data can only be moved to a remote location using "sync" methods
  - claiming back storage space is only possible with multiple deletions of files and snapshots

- **Snapshots**
  - enables you to store a point-in-time-reference of all your data (active and deleted)
  - protects you against accidental deletions and corrupt files
  - enables you to go back to a previous version of a file
  - makes a file only recoverable if a snapshot tasks has been done
  - deleted data accounts as deleted data
  - deleted data can be moved to a remote location using zfs-replication
  - recoverability depends on the snapshot deletion date

Backup solutions by popular apps

The methods Rsync, Snapshots and ZFS-Replication fit the needs of most users and are implemented well in each operating system. However some users prefer a more fine grained control. If you are such a user, you can do your own research on these popular apps:

- Restic
- Duplicacy
- Borg-Backup
- Kopia
- UrBackup
- Duplicati

Though it is not a backup solution the file synchronization software "Syncthing" is very popular and can be helpful as well.

Access via Secure Shell (SSH)

A secure shell connection enables you to access other systems from your NAS. This is a secure connection that uses authentication via passwords or special keys.

One use case are backups. Most systems will guide you through their graphical user interface to the appropriate place if you need to setup such a connection for backup purposes.

For the many other use cases you need to do individual research on your own.

Remote access

One core feature that will come in handy multiple times a day is having access to big storage capacities and all your files from multiple devices - even outside the location of the NAS. Then you are essentially your own cloud-provider but with increased data security and the option to upgrade your storage without any additional monthly fees.

To access your NAS remotely the easiest solution is to use a vpn connection. There are a lot of vpn-providers out there that enable you to connect to the web through their servers to change or hide your location. But that is not what we want.

We want a vpn connection or vpn provider that allows us to connect to our home network. For this we can setup our own vpn connection using "wireguard" or use a service like "Tailscale" or "Zerotier". Please note that a diy wireguard connection is not difficult to setup but has more dependencies for your network setup and requires more technological knowledge.

Since this is a manual for beginners here are guides to setup „Tailscale“ on different systems (click the link or scan the qr-code):

Guide written by Tailscale for installing on TrueNAS Scale

Guide written by Unraid

Alternatively you can watch these two videos if you want to setup Tailscale on TrueNAS Scale:

Step-by-step Guide: How to setup Tailscale on TrueNAS Scale

How to Configure ZFS Replication on Two or More TrueNAS Scale Systems Using Tailscale

Summary of the software concepts

Concept	Summary
Virtual devices	Multiple real drives combined to create a big storage pool
RAID	Configuration of the storage pool. There are different types, each with their own level of hardware redundancy and transfer speeds. Remember: RAID is not a Backup!
Dataset	A set or group of data that can be defined individually to your liking. Important: Datasets are the reference points for snapshots!
Backup	You can choose between full system backups vs. incremental backups. Each full system backup utilizes as much space as all data. Incremental backups only store one copy of the current data and the changes since the last backup.
Recycle bin	Protecting yourself against accidental deletions by moving files to a special folder rather than deleting it. It does not create a backup of non-deleted files.
Rsync	Synchronizing two storage locations to have one copy of the current data. You are able to restore deleted data but not previous versions of files by default but there are special commands you can use. Rsync is recommended for backing up data to a different operating system (e.g. TrueNAS Scale to Synology).
Snapshots	An incremental backup of whole datasets as a point-in-time reference for restoring data. They get stored on the same system and consume as much space as data has changed. They can restore previous versions of files as well as deleted data within the selected timeframe.
Replication	Transferring snapshots to another system that also uses the zfs-filesystem.
Secure Shell (SSH)	A secure way of accessing the NAS via the command line or from other systems.
Remote access	Via a vpn like Tailscale, Zerotier or your own wireguard connection.

It is important that you understand these concepts before you put any data on your drives. It is better to think twice before any data is lost just because you „could not wait“ or „did not have the time“.

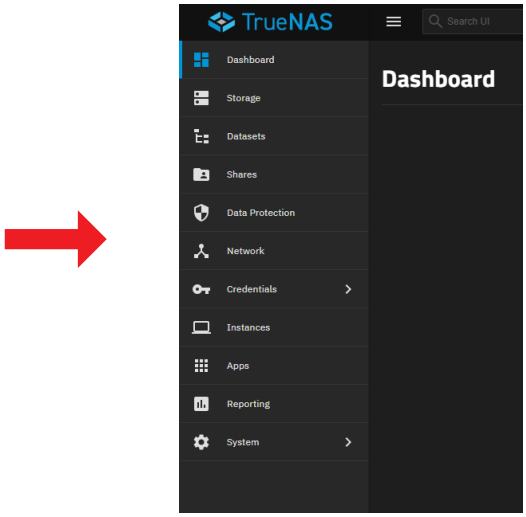
Software configuration

It is time to configure your NAS. This is an important step because you are now building the software architecture for your data. You can start over but the more data you put into your system the more difficult it gets to change basic settings. So be careful!

Please note:
The software configuration will use the words of and will be shown in the operating system "TrueNAS Scale" but the steps will be roughly the same in every operating system. You can follow guides in the documentation of each individual operating system or on YouTube to find the appropriate buttons in each individual graphical user interface (GUI).

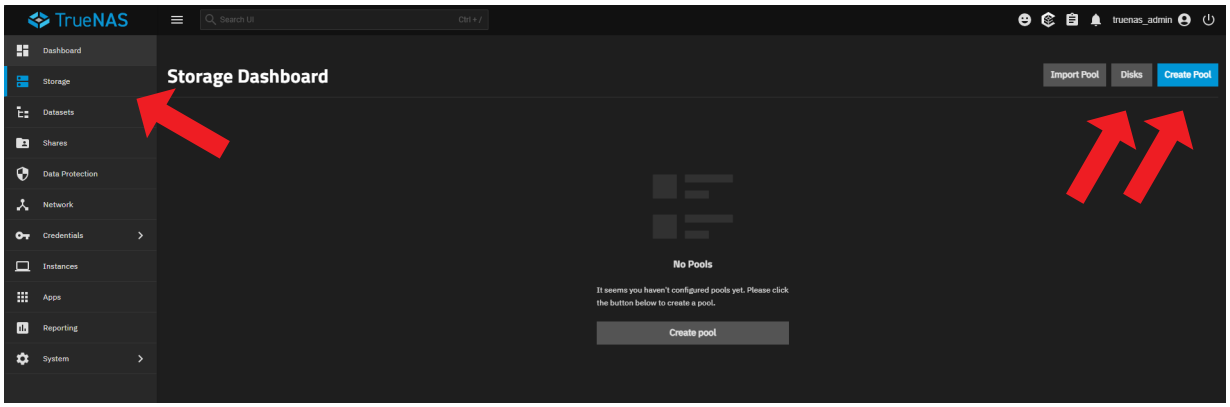
Creating a pool

On the left side of the webui you will see the following menu:

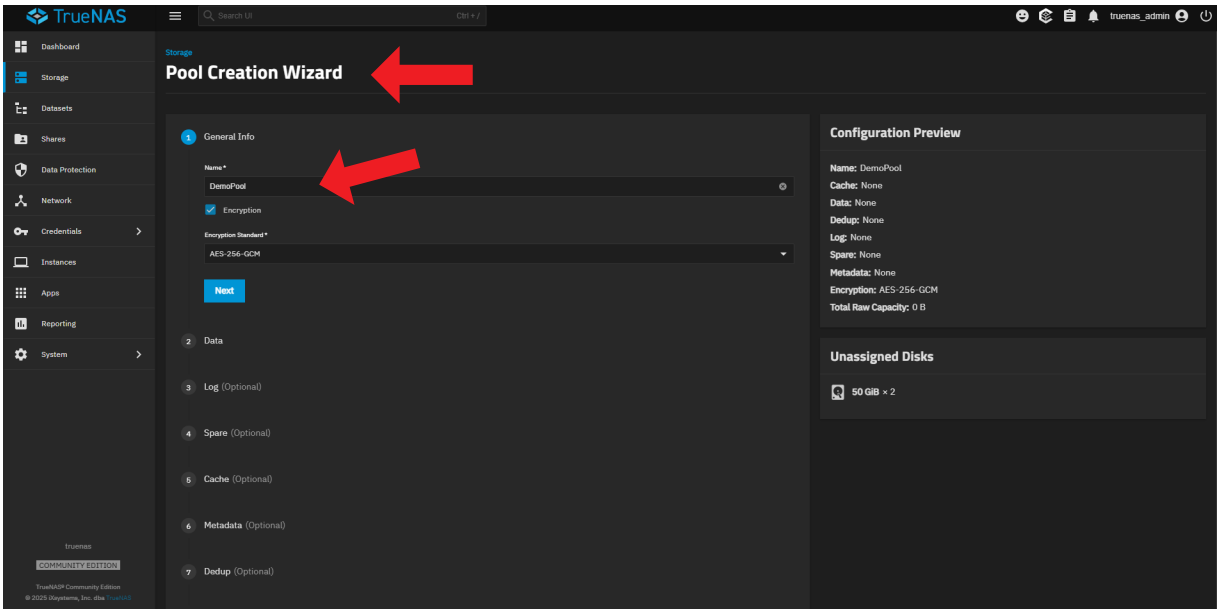


Click on „storage“ and then on „disks“ at the top right corner to check if all your drives are recognized by the system. If a disk is not showing up properly, check if the cable is correctly attached. A slight „wobble“ can cause connection issues. It is best to shutdown the system before disconnecting and connecting the drives again as most ports are not „hot plugable“.

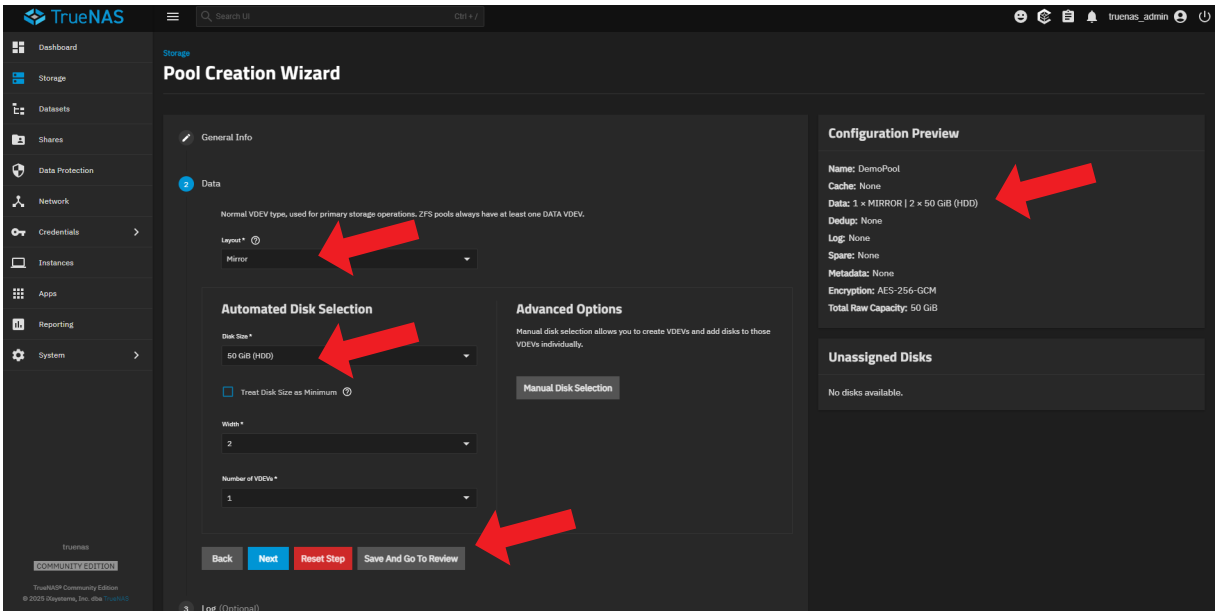
After you confirmed that all drives are connected and recongnized you can click in „create pool“.



This will start the „pool creation wizard“. In the first step you need to set a name for your pool and choose if you want to enable encryption (it is recommended to turn encryption on). Then click on „next“.

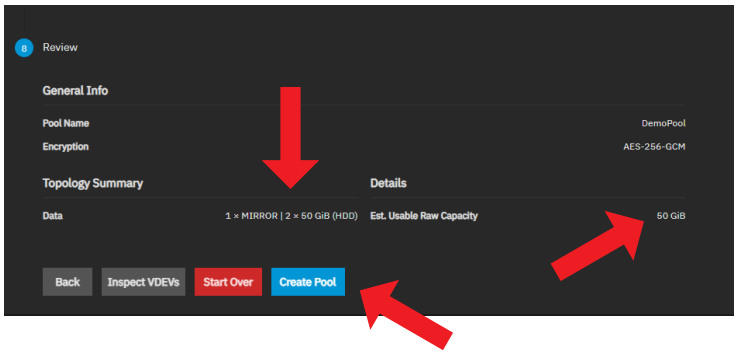


Then you need to choose your RAID configuration. In the example below a „RAID 1 (mirror) was created with two 50GB drives. From the overall 100GB of space will 50GB be usable. Choose a configuration compatible with your drives and plans you chose in the earlier chapters of this manual.

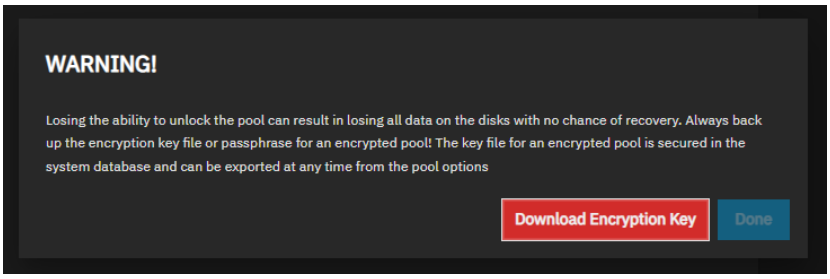


In the next steps you can configure additional drives to enable advanced features. We will skip these for now because you can always add them later to the pool. Click „save and go to review“.

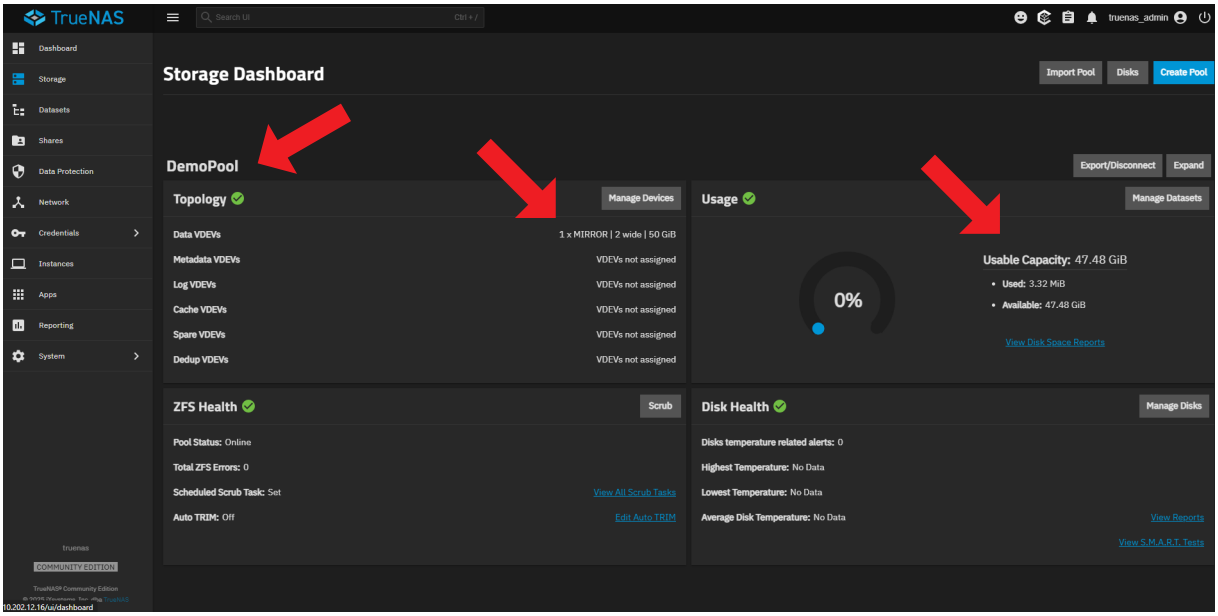
There you can see a summary of your configurations:



By clicking „create pool“ you will be prompted to „download your encryption key“ if you enabled encryption. This is the code that gets stored in the configuration of your system and that you need to unlock your data in special occasions. Download the key to a save place and click „done“:

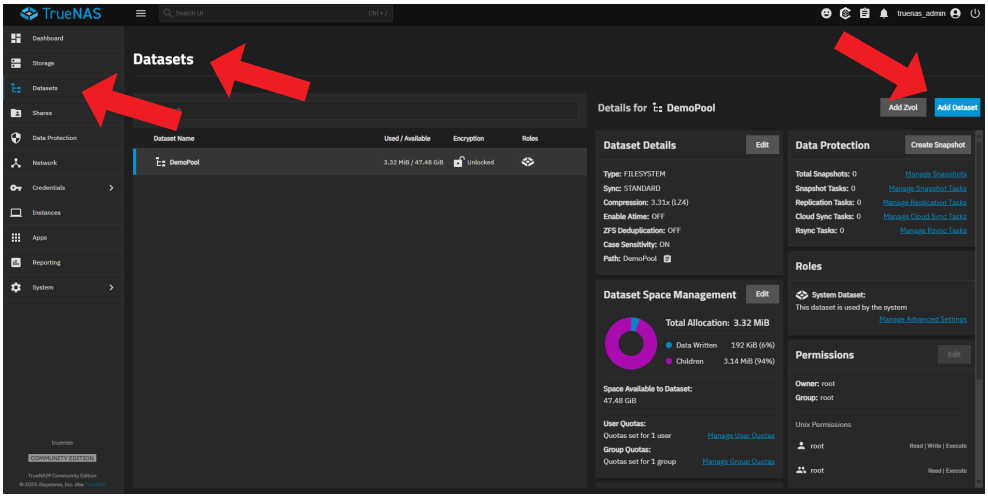

WARNING!
Losing the ability to unlock the pool can result in losing all data on the disks with no chance of recovery. Always back up the encryption key file or passphrase for an encrypted pool! The key file for an encrypted pool is secured in the system database and can be exported at any time from the pool options
Download Encryption Key
Done

Then the „pool creation wizard“ will configure your pool and you will be directed to the dashboard of your pool:

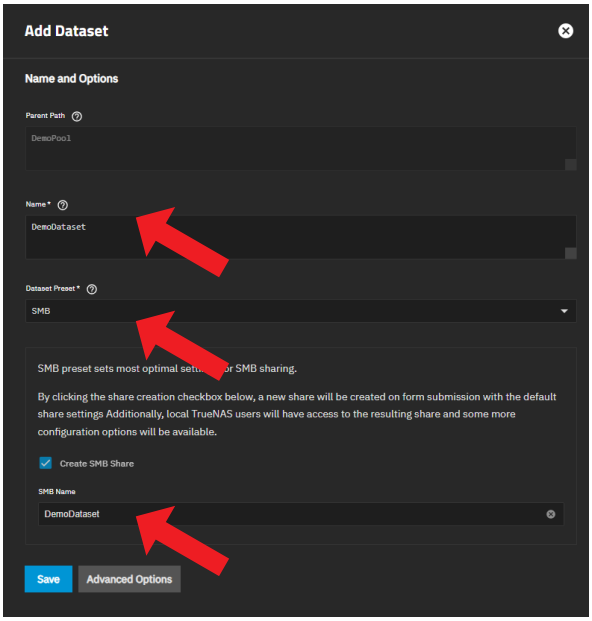


Creating datasets

By creating a pool you created a space for storing your data but you did not define a layout and destinations for snapshots. Therefore we need to define datasets based on your chosen dataset layout in earlier chapters of this manual. In the menu on the left side click „datasets“:

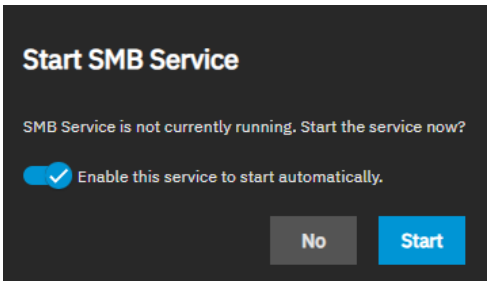


Then click „add dataset“ in the top right to get to this menu:

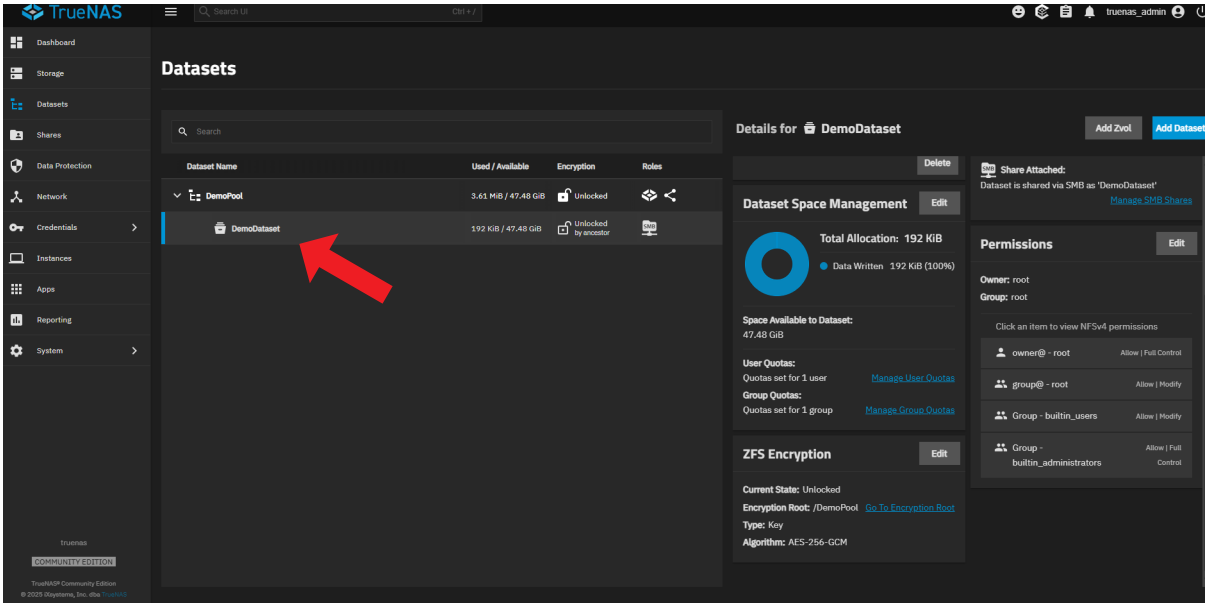

Add Dataset
Name and Options
Parent Path
DemoPool
Name *
DemoDataset
Dataset Preset *
SMB
SMB preset sets most optimal
or SMB sharing.
By clicking the share creation checkbox below, a new share will be created on form submission with the default share settings Additionally, local TrueNAS users will have access to the resulting share and some more configuration options will be available.
Create SMB Share
SMB Name
DemoDataset
Save
Advanced Options

Here you can define a name and a preset for your new dataset. Usually the „SMB“ preset is the right choice. If you choose this, a SMB share will also be created if you don’t deselect it. Click „save“.

If this is your first dataset with the smb preset you will be prompted to activate the smb service. Without this service you will not be able to access your data from other systems.


Start SMB Service
SMB Service is not currently running. Start the service now?
Enable this service to start automatically.
No
Start

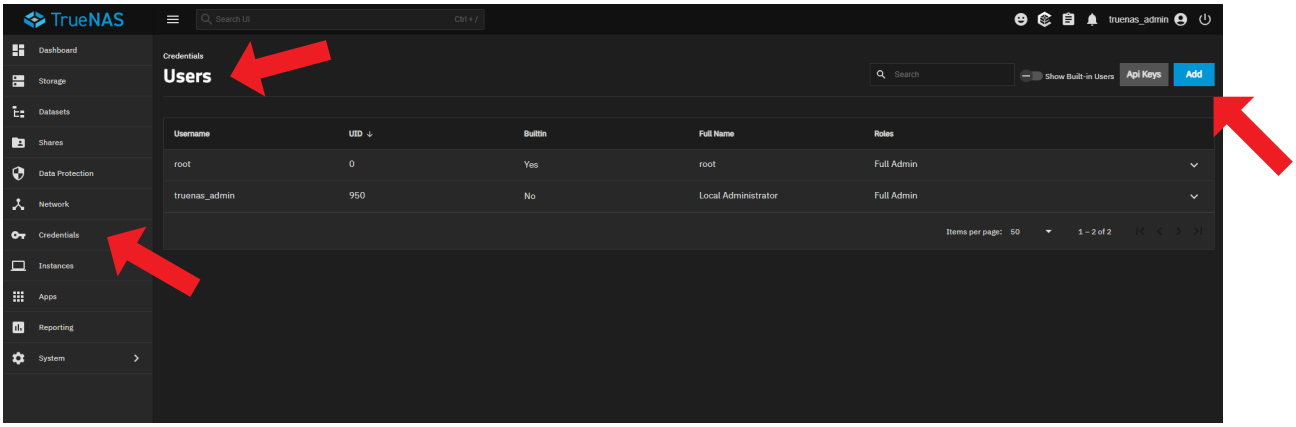
The created dataset should be listed below the pool like this:



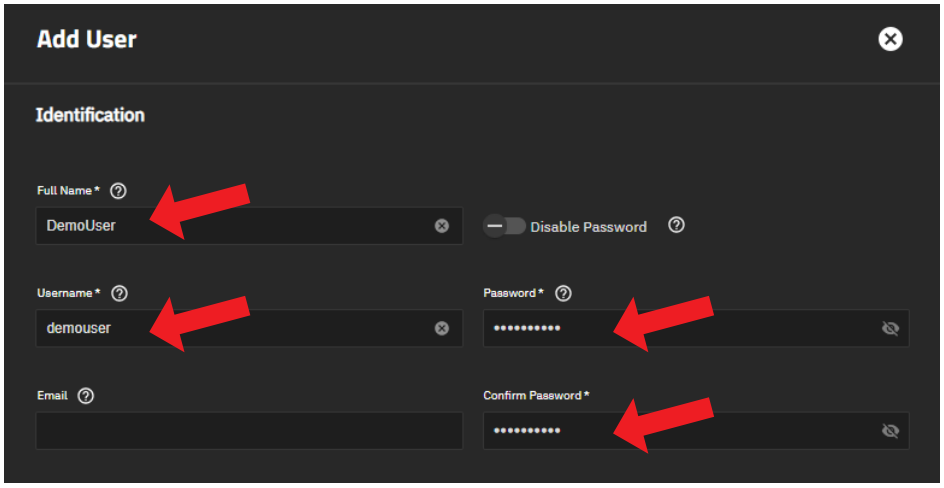
To add a new dataset you need to click on your pool (the first entry at the top) and then click „add dataset“. If you click „add dataset“ while another dataset is selected you will create a „child-dataset“ (a dataset in another dataset). Child-datasets will be included in a snapshot that is recursive.

Accessing data with smb shares (and setting permissions)

To access your data you first need to create a user. Click on „credentials“ in the left menu and then on users. You will see the following menu:



Click on „add“ in the top right corner and enter a name (username gets filled out automatically) and a password. You can check out every other option but for accessing your data you only need those two.

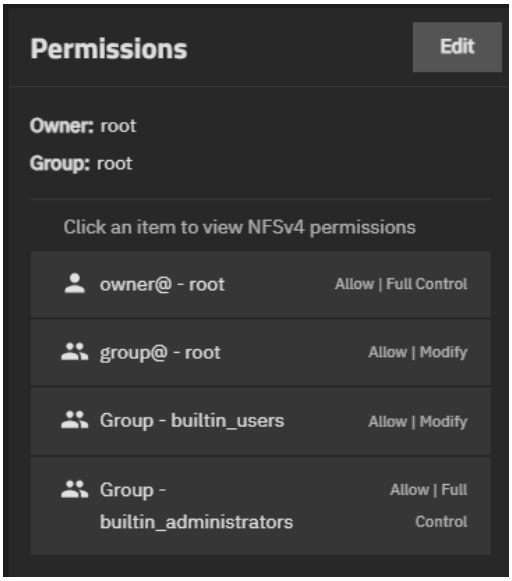


Now you can access your data from another system by following the guide for your operating system:

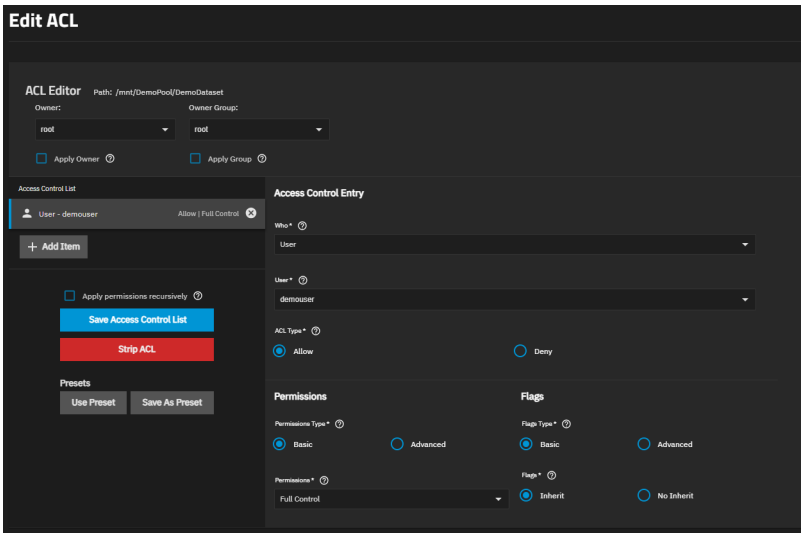
- **Windows**
  - You can permanently connect a network share:
    - Open file explorer - go to "this pc" - right click "new network share"
    - Enter „*server-name**share-name*“ in the address bar - connect
    - Enter the credentials of the user (from the NAS, not your current device)
    - The network Share will now be visible under "This PC" and you can create a shortcut to it on the desktop
  - Alternatively you can connect via one time connections:
    - Open file explorer
    - enter „*server-name**share-name*“ in the address bar
    - Enter the credentials of the user (from the NAS, not your current device)
- **macOS:**
  - Finding the NAS in the network-tab in the finder:
    - Open Finder - click on Network on the left side - click on your systems name - connect - enter credentials - save them in "Mac OS Keychain" - connect
  - "Connecting to server":
    - Open Finder and press Command + K
    - Enter „*smb://server-name/share-name*“
    - Click "Connect
    - Enter the credentials of the user (from the NAS, not your current device)
- **Linux:**
  - Most often you can find the SMB-Share under "Network" in you file browser
  - You can add an SMB-Share with „smb://server-ip/share-name“
  - Permanent mounting option:
    - Use the terminal command:

      sudo mount -t cifs //server-name/share-name /mnt/point username=user,password=pass
    - change "server-name", "share-name", "point", "user", "pass" according to your needs

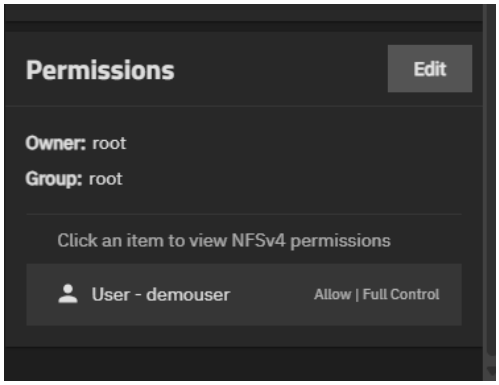
Right now every user on the NAS has access to your datasets. Most often this is not how you want to configure your NAS. To precisely define permissions you need to go back to your dataset dashboard, select the dataset you want to configure and search for „permissions“ on the right side:



Click on „edit“ to access the „ACL Menu“. Here you can add or remove users based on your needs:

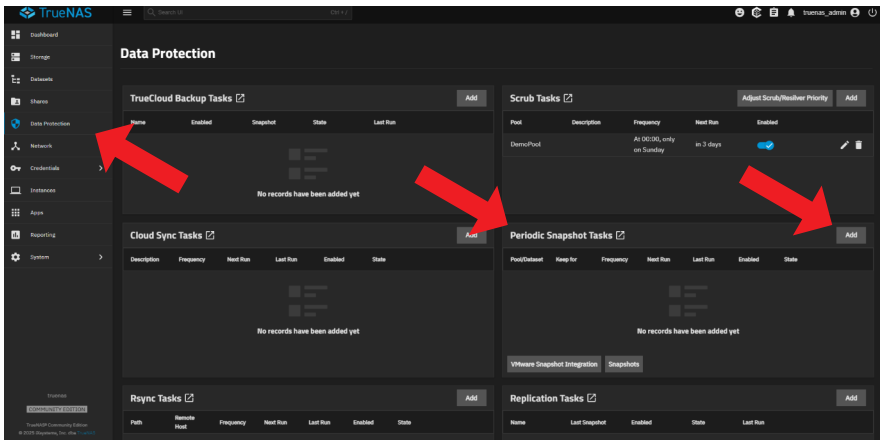


Back in the dataset settings it should look like this (for the example above):



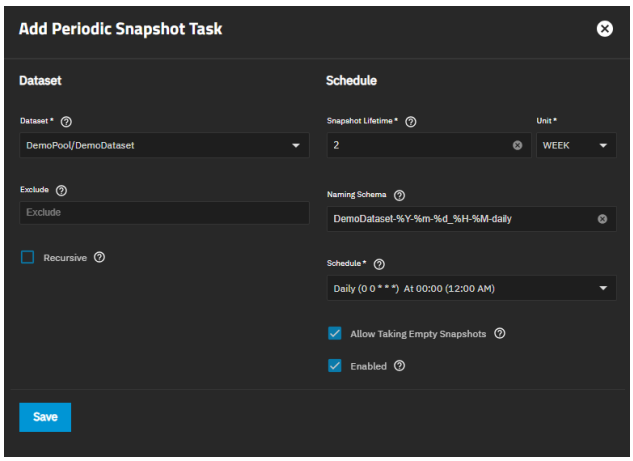
Creating snapshot tasks

To create a backup by using snapshots you need to enter the „data protection“ dashboard and search for „periodic snapshot tasks“. Click on „edit“:



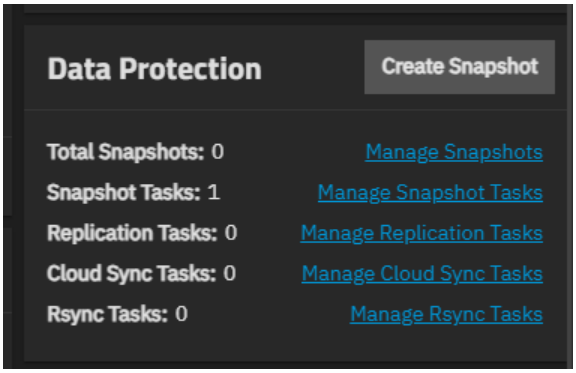
In the following menu you can choose your datasets, exclude specific or make it recursive, set a lifetime for the snapshots created, a naming scheme and a schedule.

The example below savest he „demodataset“ every day at 12AM and keeps the snapshot for two weeks. So everything that is stored for more than one day in the dataset will be recoverable fort wo weeks.

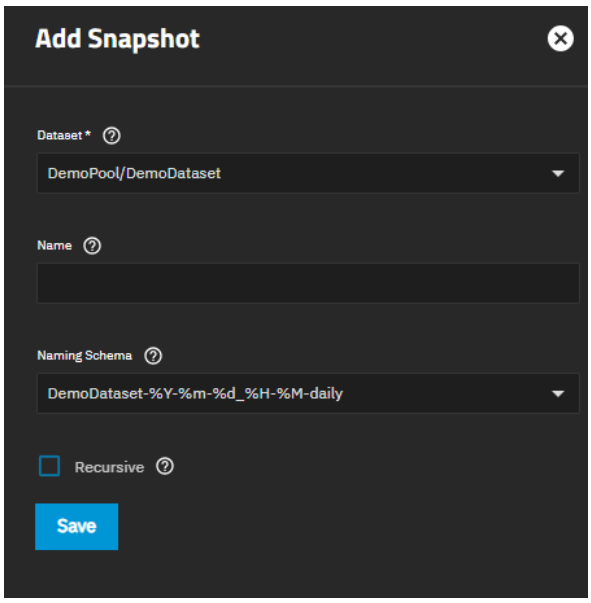


Empty snapshots should be enabled if you want to replicate those snapshots to another system.

You can check existing snapshots in the dataset menu.



You can even create custom snapshots by clicking on „create snapshot“ and filling out the following menu.



You can define a custom name or use on of your existing naming schema. By using an existing naming schema the custom snapshot will also be recognized by replication tasks.

How do I restore old data saved by a snapshot?

- TrueNAS Scale webui:
- Select the dataset and click „manage snapshots“ (on the right side)
- Select the snapshot and one of the two options:
  - Cloning: Create a copy of the snapshot to compare before restoring files
  - Rollback: Restore the whole existing dataset to a previous point in time (This will also delete every snapshot that was made after that)

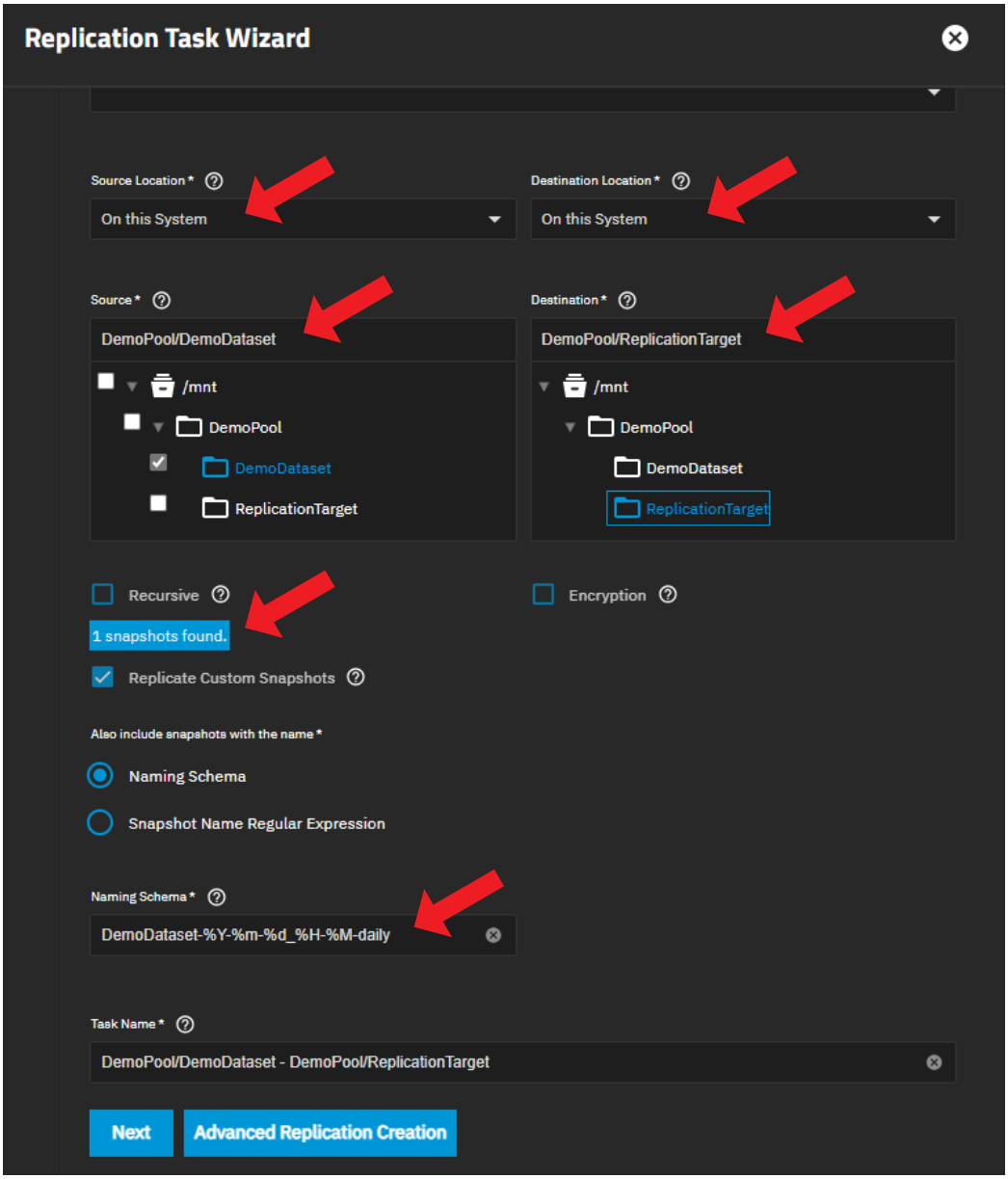
- Windows:
- Right-click on the folder you want to restore files in and select „properties“
- Select „previous versions“ and the snapshot you want to restore from
- Either select „open folder“ to copy the old files to a new destination to compare them against the files you want to replace or select „restore“ to restore the entire folder to the selected previous point in time

- Mac OS:
- The snapshot folder is hidden in every smb share individually
- Open a random folder, click „Go to“ in the top left of your screen, select „Go to folder“
- In the new window define the path like the following: „/Volumes/<*SMBShareName*>/.zfs/snapshot/“
- Browse through the snapshots and copy the files you want to restore

- Linux:
- The snapshot folder is hidden in every smb share individually
- Open the folder you want to restore files in
- Extend the path by the name of the hidden folder. The path should look like this: „smb://<*server*>/<*share*>/.zfs/snapshot/“
- Browse through the snapshots and copy the files you want to restore

Creating replication tasks

Go to „data protection“ and search for the replication task tab. Click „add“. The following example shows a configuration for a local replication from the dataset „demodataset“ to the dataset „replicationtarget“ in the same pool.



After that you just need to define the schedule of the replication task.

Replications are intended to replicate snapshots across two systems. By selecting „on another system“ for the „destination location“ you will be prompted to select an „ssh connection“. This is a secure connection between the two systems that you need to configure.

This is easier than it sounds. Just go through the prompted ssh wizard, name an admin of the remote system to login the webui of the remote system and select root for the ssh transfer. You need to enter the password for the admin of the remote system! A secure pair of keys will automatically created for you and everything is configured.

Case study

The author of this manual has built a case study to show that a NAS out of old spare computer parts is possible. This will also give you a visual representation of the end product. You can follow the build or take it as a guidance.

Use case

The NAS of the author should fulfill the following tasks:

- Be a storage server for files, photos and other media
- Be a backup solution for other devices with the ability to do remote replications
- Provide additional apps and services for enhanced network security and as a "nice to have"
- Give access to a couple of users and one or two admins
- Achieve the tasks as energy efficient as possible while "not breaking the bank"

Hardware configuration

Because one main focus point are backups the author actually has built two systems: One main NAS and one backup server for remote replications.

Based on accessibility and prices the author's hardware configurations are the following:

- **Main NAS**
  - Mainboard with eight slots for storage devices, the possibility of PCIe expansion and a one gigabit network connection
  - 2x NVMe drives (for redundant boot devices of the operating system)
  - 5x 4TB hard drives (for storing data)
  - 1x 2-core (4 threads) cpu
  - 4x 8GB RAM
  - 750W power supply (overpowered for a NAS but it was on sale)

- **Backup NAS**
  - Mainboard with eight slots for storage devices, the possibility of PCIe expansion and a one gigabit network connection
  - 1x SATA ssd (as a boot device for the operating system)
  - 1x 16TB hard drives (for storing replications)
  - 1x 2-core (4 threads) cpu
  - 2x 8GB RAM
  - 250W power supply

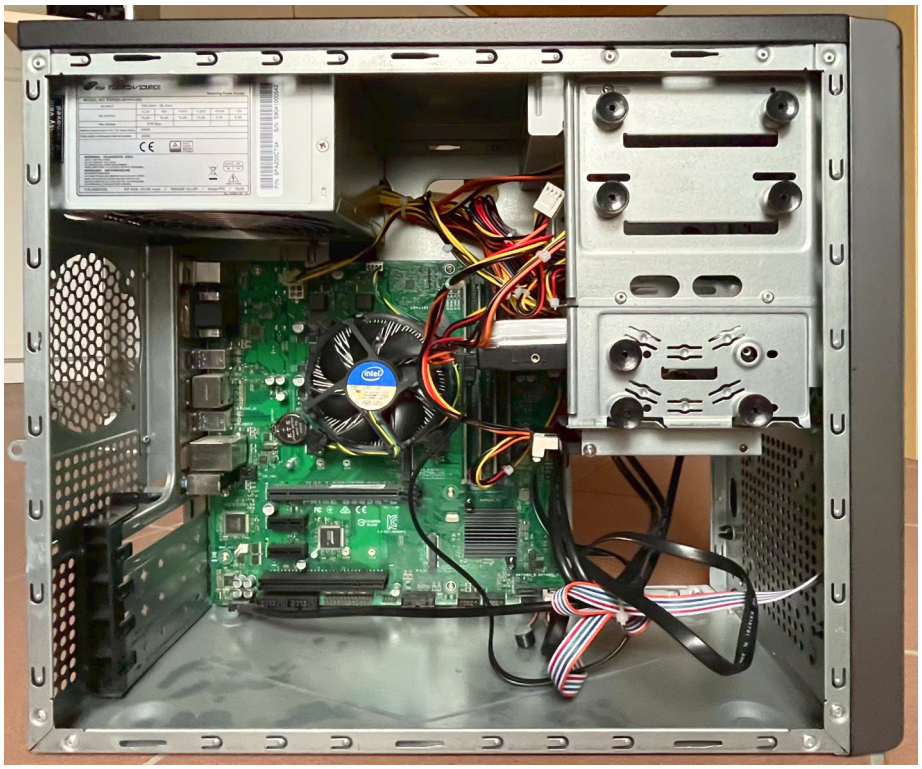
Please note that the manufacturers of the components are not mentioned on purpose because the suitability and quality of the products can vary in the future.

The backup system has a very limited hardware configuration because the whole purpose of this system is to "turn on, do replications, turn off".

Here is a visual impression of the finished builds:

Main NAS

Backup NAS



Software configuration

Regarding energy consumption the BIOS (Basic Input Output System) options of each server were checked and features like the following were enabled:

- Activate „C States“ for the cpu and allow the highest standard
- Turn off Power Enhancements for the cpu
- Turn on ERP
- Check ASPM and set it to "Auto"

The author chose the operating system „TrueNAS Scale“ and set the following settings based on his needs and use case:

- RAID6 (RAID Z2)
- Encrypted drives via keys and passphrases
- Dataset layout
    - Users
        - User 1
        - User 2
    - Media
        - Photos
        - Videos
        - etc.
    - Apps
        - App 1
        - App 2
    - BackupsFromDevices
        - User 1
        - User 2
- SMB Shares enabled as needed
- Snapshot Tasks
    - Users - Every 20 min, recursive, don't take empty snapshots, keep for 2 hours
    - Users - Every hour, recursive, don't take empty snapshots, keep for 2 days
    - Users - Every day, recursive, take empty snapshots, keep for 4 weeks
    - Media - Every day, recursive, take empty snapshots, keep for 4 weeks
    - Apps - Every day, recursive, take empty snapshots, keep for 4 weeks
    - BackupsFromDevices - Every day, recursive, take empty snapshots, keep for 2 weeks
- Replications Tasks
    - Users - Every night, keep daily snapshots as long as source
    - Media - Every night, keep daily snapshots as long as source
    - Apps - Every night, keep daily snapshots as long as source
    - BackupsFromDevices - Every night, keep daily snapshots as long as source
- No Rsync Tasks
- Remote access via Tailscale
- All security enhancements (see next chapter)

Please note:
Each app gets its own dataset because each one has its own individual needs and configuration files. If an app does crash or need to be reinstalled, you can point the installation path to the existing dataset and (if all runs well) all settings you changed should be adopted.

Replications to a remote location

The author did see no need for the backup server at the remote location to be powered on all the time since its whole purpose is to only replicate the main server. Instead he searched for a solution or process to power on the second server once a day, pull the newest snapshots from the main server and then automatically shutdown again.

Unfortunately his operating system of choice does not provide this feature by default because it is designed to be on 24/7. But with many hours researching (and a little help from AI) he found a solution: A smart power outlet, shell commands and scripts! Many tasks that can be configured via the webui can be done using shell commands and therefore be implemented in a script. Also each task has its own json-file that contains all configurations and statistics about the task. This way scripts can analyze these json-files to get information about the tasks, check status and run tasks via scripts.

You can always see the current version of his method here: Github

Conclusion

Here are some statistics about the case study:

- **Main NAS**
  - 12 TB of usable space
  - RAID6 (RAID Z2)
  - Snapshots secure important data every 20 min, hourly and daily
  - Additional apps and services are provided to the network
  - Remote access is enabled
  - Power consumption: 35 Watts (24/7, all disks spinning all the time)

- **Backup NAS**
  - 16TB for backups
  - RAID0 („Stripe“)
  - Fully automated remote replications
  - Power consumption: 26 Watts (roughly 30 min/day)

- **Total power consumption:**
  - 311.35 kWh/year (at 0,30/kWh = 94/year = 7,80/month)

Wish list (as of writing this manual):

- Improving the backup scripts
- Lower the energy consumption
- Easier / more fine-grained control of backing up individual clients

Additional considerations

Security enhancements

- Protect shell access with an admin password
  - The shell is a a command line interface that you can see when you attach a display to the NAS. You can enter commands to change the system settings.
- Encrypt your pools and datasets
  - protects your data against theft
  - a passphrase is more secure than a code (the code gets saved in the config of the system and unlocks your dataset by starting the system, a passphrase does not)
- Enable two-factor-authentication for login
- Only allow access to only what is needed
- Keep your system up-to-date
- Disable unused servcies
  - All services like "SMB" and "SSH" are disabled from the start
  - Disable activated services that are not in use
- Limit physical access to the NAS
- Save your configuration regularly
  - Configuration of the NAS are often saved to a special file. Backing up this file is recommended
  - This is your backup of all the settings you make in systems like TrueNAS Scale. This does not get backed up automatically by snapshots!

Please note:
After enabling two factor authentication don't close the current tab of your browser before opening up another tab and confirming that you can login using 2FA. You can disable 2FA in the current tab (where you are still logged in) but if you don't have access to that tab and can't login, you successfully locked yourself out of your own system. The only option is to reinstall the operating system and importing and unlocking your pool.

Setting up alerts

Operating systems like „TrueNAS Scale" can alert you via mail. You need a mail account to receive mails and one to send mails. The sending mail account needs to be accessible via „remote settings". In „TrueNAS Scale" you can enable mail alerts by like so:

- Click on „system settings" and „general settings"
- Scroll down to „Mail" and add your mail address
- Click the alert icon at the top right of the webui and go to „alert settings"
- At the top configure a new alert setting
- Confirm mail notifications via a test mail
- Define your alert settings for each event in the alert settings menu

Creating a „plan for disaster“

In IT there are "technical and organizational measures" to protect systems. These help to create plans for all kinds of scenarios so everyone knows what to do if something bad happens.

It is something that can be scary but you will thank yourself later if you think about these things now. Here is a list of what you can define:

- Define "What if"-Scenarios and Emergency Plans
- What will you do in case of a fire, water damage, theft, corrupted or deleted files?
- How often and where is the data still accessible?
- How quickly can a backup be performed?
- Test your backup recovery options regularly!

Documentation

Most people are fascinated about something for a limited time, want to know everything about it and then… they don’t. They loose interest, they get used to something and don’t think about it anymore.

This can happen with you and your NAS, too. Protecting you against yourself can be beneficial by documenting what you did. The best tip the author can give you: Document everything!

- Why did you chose your hardware components and operating system?
- Why did you configure your software the way you did?
- How do you need to configure another user or an app to fit your needs?
- and so on....

You can use any tool you like - word, excel or even a graphical representation with Drawio.

In your documentation you could also note future plans like a wish list of features or hardware options.

By noting things down you don't have to do the same work over an over again. And you don’t break things that you did for a specific reason because you „didn’t know why you did something“.

Final words

YOU DID IT!

You now have the knowledge to build your own NAS and a place to store all your data in a safe, consistent and regularly backed up manner. **Well done!**

You've not only assembled a powerful piece of technology, but you've also laid the foundation for a valuable tool that will serve you for years to come.

However, remember that the journey doesn't end once your NAS is built. It's essential to consider how you'll integrate it into your daily routine - whether it’s for storing files, backing up data, or sharing across devices, make sure it fits seamlessly into your life.

Regular maintenance is key: Check on your NAS periodically to ensure everything is working as expected, backups are up-to-date, and no errors have cropped up.

A NAS is another device to regularly maintain but it makes sure that all of your other devices data is safe!

Thank you for trusting this guide to help you through the process - your DIY-NAS is now ready to provide you with reliable storage and support. I appreciate your dedication, and wish you the best in your new, self-made digital setup!

Designed by storyset at Freepik.com

Discover this great guide for building your own DIY Network-Attached Storage (NAS) system! Perfect for tech enthusiasts and anyone seeking better data security, this manual walks you through selecting hardware—whether new or repurposed—and configuring software step by step. Packed with essential tips on backups, data protection, and insightful questions to tailor your build, it empowers you to take control of your digital life. With clear instructions and practical advice, this book makes NAS building both approachable and rewarding!

„The one who now has a bullet-proof backup system“, 2025

Designed by Freepik

Created in 2025 by SturmFlosseTK

First published at Internet Archive

(CC BY-NC-SA)